Universidad Carlos III de Madrid

Escuela Politécnica Superior

PROYECTO FIN DE CARRERA

Ingeniería en Telecomunicación

CALUS

Sistema de ayuda a la evaluación de asignaturas enmarcadas en el Espacio Europeo de Educación Superior

Aplicación a manejo de herramientas software y equipos hardware en asignaturas de base tecnológica

Autor: Cristóbal Miranda Puente
Tutora: Celia López Ongil

Agradecimientos

Hace ya algunos años que empecé la carrera, un poco perdido y asustado como todos, pero con una ilusión inmensa por descubrir y aprender algo nuevo. Ni un sólo día ha pasado desde entonces, sin que me haya alegrado de la decisión que tomé y que algunos me aconsejaron tomar. Gracias papá.

Ahora ya parece que se acaba lo que empezó como un sueño. Pero sólo lo parece, porque una etapa así nunca puede terminar. Muchísimo tengo que agradecer y a muchísimas personas que nombrar en estas paginas, que ya estaban escritas desde antes de empezar este proyecto.

En primer lugar a mi familia. A mis padres por el enorme sacrificio que han hecho, no sólo en esta etapa sino desde que nací. Por su amor incondicional y por todo lo que han cuidado de mí desde entonces.

Y por supuesto a mis hermanos. A Dani por haber compartido conmigo los momentos más importantes de mi vida y por haber sido además de hermano, amigo y compañero. Y a Javi, la mejor persona que he podido conocer hasta hoy, y que a pesar de la diferencia de edad siempre ha estado con nosotros.

En segundo lugar a todos los amigos que he tenido la suerte de encontrar en el camino, Javi, Fer, Iñigo, Relan, Gutsi, Carlitos, Paloma, Maria, Miguel, Natalia, Fernando, Jose y muchos más (que nadie se ofenda si no he escrito su nombre, porque me acuerdo de cada uno de vosotros). Porque sin ellos no habría podido finalizar está dura prueba. Cada clase, cada práctica, cada entrega, cada examen. Por todas las risas y lágrimas. Por todos los apuntes que me habéis dejado, por las comidas en el comedor, por aquellas horas de minijuegos con las que tanto nos hemos divertido, por todas las fiestas que hemos disfrutado y sobre todo por ese increíble viaje que hicimos. Gracias a todos.

Una mención especial a mi grandísimo amigo Adrián, la persona más autentica que he tenido el placer de conocer y que ha sufrido conmigo mil y una prácticas desde que empezamos. Y por supuesto a Pablo e Isabel, por todo lo que he aprendido con ellos, y sin cuya ayuda nunca habría aprobado Microondas.

En tercer lugar a todos mis profesores, por dedicar su tiempo a una labor tan especial y dura. Por entregarme sus conocimientos y enseñarme mucho más que una carrera. Gracias a Celia por ofrecerme el proyecto y confiar en mi incondicionalmente. Por todas las mañanas y tardes que hemos pasado hablando y aprendiendo. Por toda su ayuda y su entrega, y por todo el tiempo que nos ha dedicado al proyecto y a mí. Además, nombrar a Mario, Marta y Susana, que aún no teniendo que ver con el proyecto me han resuelto innumerables dudas y problemas.

En cuarto lugar a mis amigos, Lorenita, Vane, Chabe, May, Jorge, Adri y por supuesto Elena y Dani. Nunca olvidaré el día que os conocí a cada uno de vosotros. Hemos compartido tantos buenos momentos que parece imposible haberlos vivido todos en tan poco tiempo. Gracias por ser como sois.

Y por último, a la persona más importante de todas, y no sólo dentro de este proyecto, Elena. Sin ella nada de esto hubiera sido posible. Nunca podré agradecerte todo el

apoyo y el esfuerzo que has hecho para ayudarme a conseguirlo. Gracias por estar ahí en cada triunfo y en cada tropiezo, por estar siempre pendiente de cada detalle, por escucharme y aconsejarme en cada decisión y por aguantarme en los peores momentos. Y sobre todo, gracias por haber decido compartir tu vida conmigo y por quererme tanto. Te quiero.

En definitiva, a sido la experiencia más intensa que he vivido hasta este momento. Espero que haya muchas más en la vida que se me presenta a partir de ahora, pero que aún así, nunca cambiarán lo vivido.

Índice

Índice de figuras

Índice de tablas

1. Introducción

1.1 Ámbito general del problema y objetivos

El 19 de junio de 1999, los ministros de educación de 29 países europeos firman la Declaración de Bolonia, que da el nombre al proceso, con el objetivo de desarrollar el Espacio Europeo de Educación Superior (EEES) antes del año 2010 [1].

La principal reforma consiste en crear un Espacio Europeo de Educación Superior competitivo y que sea atractivo tanto para estudiantes y docentes como para terceros países.

Los cambios más sustanciales que se van a producir se pueden sintetizar en tres grandes grupos: las adaptaciones curriculares, las adaptaciones tecnológicas y las reformas financieras necesarias para crear una sociedad del conocimiento.

El proceso de Bolonia propone la creación de un EEES cuyos objetivos fundamentales son [18]:

1. **Adopción de un sistema comparable de titulaciones –** No implica que las nuevas titulaciones sean las mismas para todos los países firmantes, sino que se basa en el reconocimiento de titulación y no de conocimientos.
2. **Adopción de un sistema basado en tres ciclos (grado, máster y doctorado) –** Se pretende conseguir mejor incorporación de los estudiantes al mundo del trabajo.
3. **Establecimiento de créditos ECTS (European Credit Transfer System) –** Es un sistema de transferencia de créditos que cuentan no sólo las horas de clases teóricas (es decir, las impartidas por el profesor y las horas de examen) sino también el trabajo que debe ser realizado por el alumno (seminarios, horas de estudio, realización de trabajos). El crédito ECTS corresponde a entre unas 25 y 30 horas.

El EEES implica la instauración de nuevas metodologías docentes, en detrimento de las tradicionales clases magistrales. Para ello propone dos nuevas metodologías:

- **Evaluación continua** – Seguimiento diario al trabajo personal del alumno mediante evaluaciones continuas. Para llevar a cabo la evaluación continua se proponen principalmente dos herramientas:
  - Uso de todas las posibilidades que ofrece Internet y las nuevas tecnologías TIC
  - Tutorías personales.
- **Enseñanza práctica** – Intervención activa del alumno a través de ejercicios, trabajo en grupo, prácticas profesionales, etc.

En España este proceso se ha retrasado respecto a la mayor parte de Europa. Hasta Octubre de 2007 no se promulgó la normativa de los nuevos ciclos universitarios, por lo que las universidades tienen sólo los cursos 2008//09 y 2009/10 para adaptar sus estudios al marco común europeo.

En la Universidad Carlos III de Madrid, la reforma de Bolonia ha entrado en funcionamiento en el año 2008/2009. Ha sido una de las primeras universidades españolas en afrontar este nuevo cambio.

Uno de los principales problemas que se plantean, es la medida objetiva del trabajo de cada uno de los estudiantes. Debido a los ya mencionados créditos ECTS, es necesario llevar un control del trabajo tanto individual como colectivo de los estudiantes.

Para poder gestionar y controlar la evaluación de estos créditos, va a ser necesario el desarrollo de nuevas aplicaciones que automaticen y faciliten la adaptación de las antiguas evaluaciones de los estudiantes.

Este es el principal objetivo de este proyecto, con el que se desea generar y gestionar un conjunto de estadísticas que permitan llevar cabo estas nuevas evaluaciones con la mayor cantidad de datos posibles.

Dado que en la Escuela Politécnica Superior de la universidad Carlos III de Madrid, gran parte del trabajo de los estudiantes esta dedicado a las actividades en los laboratorios y aulas informáticas, este sistema ha sido implementado para llevar a cabo un control minucioso del trabajo de los estudiantes en dichos entornos.

Además, proporciona un mecanismo de control de acceso basado en autenticación, para gestionar automáticamente el acceso a áreas sensibles como lo son los laboratorios y las aulas informáticas.

En la actualidad el acceso controlado a las aulas esta muy reducido, siendo muy pocas las que cuentan con un sistema de autenticación. Adicionalmente, los sistemas de acceso a los equipos y a las aulas están desvinculados, lo que imposibilita una gestión centralizada de todos los accesos independientemente de su naturaleza.

Para resolver estos problemas e incorporar nuevas mejoras, en el departamento de Tecnología Electrónica de la Universidad Carlos III de Madrid, se está considerando la posibilidad de instalar un sistema autónomo de seguimiento del trabajo personal del alumno, y es en este marco en el que se ha desarrollado este proyecto Fin de Carrera "CALUS".

Los principales objetivos de CALUS son:

- Establecer un sistema de validación que restrinja el acceso tanto a los laboratorios y aulas como a los puestos informáticos, permitiendo su uso a los siguientes colectivos universitarios: Alumnos, Personal de Administración y Servicios (PAS) y Personal Docente e Investigador (PDI).
- Eliminar la dependencia directa con bases de datos remotas, para evitar problemas de acceso durante caídas de la red, ya sean controladas o inesperadas.
- Mantener un registro de sesiones que permita conocer que usuarios accedieron a la sala y quienes iniciaron una sesión en un equipo, además de las fechas y horas de inicio y fin de sesión.
- Sentar las bases para la posibilidad de implantar sistemas de autoevaluación para los alumnos en los puestos de laboratorio

Con esta información podrá conocerse la actividad de cada uno de los usuarios y llevar a cabo una evaluación de la misma referente a los créditos ECTS.

Para la realización de este proyecto fin de carrera se ha considerado la actividad de los alumnos en los laboratorios docentes relacionados con las asignaturas del departamento de Tecnología Electrónica, pero en general es aplicable a cualquier departamento de la Escuela Politécnica Superior. Dichas asignaturas cubren un amplio abanico de herramientas con las que el alumno debe familiarizarse para su posterior aplicación en su carrera profesional. La experiencia en las mismas debe ser fruto de un esfuerzo personal y constante por parte del alumno. El nuevo marco de educación que se está implantando en el sistema europeo impone una pequeña guía por parte del profesorado y una gran cantidad de trabajo individual por parte del alumno. Sólo a través de un seguimiento objetivo y una evaluación continua parece razonable evaluar el grado de habilidad adquirido por el alumno en las herramientas, que le servirán como base fundamental de su futura labor como ingeniero o licenciado. Por supuesto, es fundamental realizar evaluaciones adicionales sobre resolución de problemas y/o trabajos utilizando estas herramientas. Este proyecto fin de carrera establece las bases necesarias para ayudar a los profesores en este nuevo reto de la evaluación continua en el Espacio Europeo de Educación Superior.

Las mejoras que implementa el sistema CALUS son:

- Supresión de sistemas de control de acceso complejos, ya que todo el sistema está basado en la utilización de una memoria flash accesible mediante interfaz USB (llave USB) que permite de forma rápida, barata y fiable el acceso a grandes cantidades de información relativas a los usuarios.
- Facilidad en la creación de nuevas llaves o repuestos de anteriores.
- Estadísticas de utilización de los equipos y de las herramientas por parte de los usuarios.
- Estadísticas históricas, como el número de sesiones iniciadas y número de accesos a las aulas agrupadas por edificio, Campus y/o toda la universidad.

1.2 Estructura del proyecto

Este proyecto se ha estructurado en 12 capítulos que se resumen a continuación:

- En el primer capítulo (Introducción) se muestra una visión general del problema que pretende resolver el presente proyecto y los objetivos que persigue.
- En el segundo capítulo (Estado de la técnica) se detallan las tecnologías que se han utilizado para la realización del proyecto.
- En el tercer capítulo (Entorno de trabajo) se describen las herramientas utilizadas para el desarrollo del sistema.
- En el cuarto capítulo (Descripción del sistema) se describen la especificación del sistema.
- En el quinto capítulo (Diseño y desarrollo) se detalla el diseño de cada uno de los módulos, el desarrollo que se ha llevado a cabo y el funcionamiento de cada uno de ellos.
- En el sexto capítulo (Planificación de la instalación) se describe el proceso de instalación de cada uno de los subsistemas en los laboratorios o aulas informáticas de la universidad.
- En el séptimo capítulo (Análisis del sistema) se muestran los resultados de las pruebas llevadas a cabo por diversos usuarios.
- En el octavo capítulo (Conclusiones) se describen las conclusiones a las que se han llegado tras la realización de este proyecto.
- En el noveno capítulo (Líneas de trabajo futuro) se proponen posibles mejoras y ampliaciones.
- En el décimo capítulo (Referencias) se resume la bibliografía utilizada para la elaboración de este proyecto.
- En el décimo primer capítulo (Glosario) se incluye un glosario con los términos utilizados en la memoria.
- En el décimo segundo capítulo (Anexos) se incluyen varios documentos que detallan algunos de los aspectos más importantes de determinadas partes de la implementación del sistema.

2. Estado de la técnica

En esta sección se hará una descripción de las tecnologías existentes que tienen relación con este proyecto.

En primer lugar se hablará sobre los sistemas de control de acceso, sus características y funcionamiento. En segundo lugar se describirán los aspectos más relevantes de la interfaz USB (Universal Serial Bus), y la evolución que llevo consigo en las comunicaciones entre los distintos tipos de periféricos, ordenadores y equipos de procesamiento. En tercer lugar se hará una introducción a la tecnología de almacenamiento de estado sólido y la interfaz de comunicaciones USB. Finalmente se hará una introducción a la criptografía y a algunos conceptos sobre este tema.

2.1. Sistemas de Control de Acceso

Desde prácticamente los inicios de la humanidad siempre se ha tenido un interés especial por los sistemas de control de acceso a distintos tipos de zonas restringidas. Al principio de forma más rudimentario y cada vez con una presencia mayor de las nuevas tecnologías y de la electrónica más avanzada se ha ido abriendo paso este campo de la seguridad.

Pero sin ninguna duda, la llegada de la sociedad de la información ha fomentado la aparición de nuevos sistemas cada vez más sofisticados e inteligentes para restringir el acceso controlado a diferentes recursos, ya sean físicos o lógicos, a determinados sectores humanos.

El control de acceso es la habilidad para permitir o denegar el acceso a un recurso particular por una entidad particular. Los mecanismos de control de acceso se pueden usar para gestionar recursos físicos (como acceso a la red de metro, donde sólo los usuarios con el ticket pueden pasar), recursos lógicos (como las cuentas bancarias, con un número de personas autorizadas), o recursos digitales (como por ejemplo, documentos de texto privados dentro de un ordenador, que sólo ciertos usuarios pueden leer).

2.1.1. Seguridad física

En seguridad física, el término control de acceso hace referencia a la práctica de restringir la entrada a una propiedad, edificio, o sala a personal autorizado. Los sistemas de control de acceso físicos pueden ser llevados a cabo por humanos (un guardia, porteros o recepcionistas) o por sistemas mecánicos como cerrojos o cerraduras, pasando por sistemas electrónicos como puertas antirrobo. En estos entornos, el uso de llaves físicas es utilizado como un medio para gestionar y monitorizar los accesos a áreas con accesos mecánicos.

El problema del control de acceso físico se puede concretar en tres preguntas: quién, dónde y cuándo. Un sistema de control de acceso determina quién tiene permiso para entrar o salir, dónde puede entrar o salir, y cuándo puede entrar o salir. Históricamente esto se ha venido solucionado mediante llaves y cerrojos. Cuando una puerta está cerrada alguien con la llave correspondiente puede entrar dependiendo de cómo esté

configurado el cerrojo. Los cerrojos mecánicos y las llaves no permiten restricciones temporales de acceso, así como tampoco proporcionan estadísticas de uso de una determinada puerta. Además las llaves se pueden copiar de una forma más o menos sencilla o transferirlas a personal no autorizado. Cuando se pierde una llave o el poseedor de ella ya no está autorizado a entrar en la zona restringida, es necesario cambiar todo el sistema.

Los sistemas electrónicos de control de acceso utilizan microprocesadores para solucionar todas estas limitaciones de las llaves y cerrojos mecánicos. Los sistemas electrónicos garantizan el control basado en las credenciales presentados. Cuando se concede el acceso, la puerta se desbloquea por un tiempo determinado y la transacción es almacenada digitalmente. Cuando se rechaza el acceso, la puerta permanece cerrada y evento es almacenado. El sistema también monitorizará la puerta y hará saltar la alarma si dicha puerta es forzada o mantenida abierta durante mucho tiempo.

2.1.2. Seguridad digital o de equipos electrónicos

En seguridad de ordenadores, el control de acceso incluye autenticación, autorización y auditoría. También incluye otros temas como dispositivos físicos, incluyendo escáneres biométricos y cerrojos metálicos, firmas digitales, encriptación, barreras sociales, y monitorización por humanos y por sistemas automáticos.

En cualquier modelo de control de acceso, las entidades que pueden llevar a cabo acciones en el sistema se denominan sujetos, y las entidades que representan recursos a los cuales el acceso necesita ser controlado se denominan objetos. Sujetos y objetos se deben considerar como entidades software, más que como usuarios humanos, ya que un humano sólo puede llevar a cabo acciones sobre el sistema a través de las entidades software que controlan.

Los sistemas de control de acceso proporcionan los servicios principales de identificación y autenticación (I&A), autorización y responsabilidad, donde:

- La identificación y la autenticación determinan quién puede acceder a un sistema, y la asociación entre los usuarios y los sujetos software, que estos pueden controlar como resultado del acceso al sistema.
- La autorización determina qué puede hacer un sujeto.
- La responsabilidad identifica qué es lo que ha hecho un sujeto (o todos los sujetos relacionados con un usuario).

2.2. Universal Serial Bus (USB)

Desde la aparición de la interfaz de comunicaciones en 1996, el protocolo de comunicación USB (Universal Serial Bus) se ha convertido en el estándar de transferencia de información entre dispositivos y ordenadores o equipos de procesamiento, desplazando a los anteriores estándares del puerto serie y paralelo. [19]

2.2.1. Historia

Antes de la aparición del protocolo de comunicación serie USB, existían ya protocolos serie o paralelo, pero surgió la necesidad de unificar todos los conectores creando uno más sencillo y de mayores prestaciones. Así nació el USB (Universal Serial Bus), una nueva arquitectura de bus o un nuevo tipo de bus que además de incrementar la velocidad de la comunicación, también permitía la conexión de varios dispositivos en un sólo puerto.

El protocolo USB fue creado en 1996 por un conjunto de siete empresas: IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC.

2.2.2. Especificaciones

El Universal Serial Bus (bus universal en serie) es un protocolo cuya función es facilitar la interconexión de distintos tipos de periféricos con ordenadores o equipos de computación.

El estándar incluye la transmisión de energía eléctrica al dispositivo conectado. Algunos dispositivos requieren una potencia mínima, así que se pueden conectar varios a un concentrador sin necesitar fuentes de alimentación extra. La gran mayoría de los concentradores incluyen fuentes de alimentación que brindan energía a los dispositivos conectados a ellos, pero algunos dispositivos consumen tanta energía que necesitan su propia fuente de alimentación. Los concentradores con fuente de alimentación pueden proporcionar corriente eléctrica a otros dispositivos sin quitarle corriente al resto de la conexión (dentro de ciertos límites).

Uno de los objetivos del diseño del USB era eliminar la necesidad de adquirir tarjetas separadas para poner en los puertos bus ISA o PCI, y mejorar las capacidades *plug-and-play* y *hot-pluggable*, permitiendo a estos dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciarlo. Cuando se conecta un nuevo dispositivo, el servidor lo enumera y agrega el software necesario para que pueda funcionar.

El USB puede conectar periféricos como ratones, teclados, escáneres, cámaras digitales, teléfonos móviles, reproductores multimedia, impresoras, discos duros externos, tarjetas de sonido, sistemas de adquisición de datos y componentes de red. Para dispositivos multimedia como escáneres y cámaras digitales, el USB se ha convertido en el método estándar de conexión. Para impresoras, el USB ha crecido tanto en popularidad que ha desplazado a un segundo plano a los puertos paralelos, ya que hace mucho más sencillo poder agregar más de una impresora a un equipo.

En el caso de los discos duros, es poco probable que el USB reemplace completamente a los buses (bus ATA (IDE) y bus SCSI), pues el protocolo USB tiene un rendimiento más lento que estos otros protocolos estándares. Sin embargo, el USB tiene una importante ventaja en su habilidad de poder instalar y desinstalar dispositivos sin tener que abrir el sistema, lo cual es útil para dispositivos de almacenamiento externo.

El USB casi ha reemplazado completamente a los teclados y ratones PS/2, hasta el punto de que un amplio número de placas base modernas carecen de dicho puerto o solamente cuentan con uno válido para los dos periféricos.

2.2.3. Velocidad en las conexiones USB

Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

- Baja velocidad (1.0): Tasa de transferencia de hasta 1.5 Mbps (192 KB/s). Utilizado en su mayor parte por dispositivos de interfaz humana (Human interface device, en inglés) como los teclados, los ratones y los joysticks.
- Velocidad completa (1.1): Tasa de transferencia de hasta 12 Mbps (1.5 MB/s). Ésta fue la más rápida antes de la especificación USB 2.0, y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad. Estos dispositivos dividen el ancho de banda de la conexión USB entre ellos, basados en un algoritmo de búferes FIFO.
- Alta velocidad (2.0): Tasa de transferencia de hasta 480 Mbps (60 MB/s). Es el estándar que se utiliza en la actualidad. Se utiliza principalmente para dispositivos de almacenamiento masivo como discos duros de alta capacidad.
- Súper alta velocidad (3.0): Se está implantando en la actualidad. Tiene una tasa de transferencia de hasta 4.8 Gbps (600 MB/s). La velocidad del bus es diez veces más rápida que la del USB 2.0 (*véase sección 2.2.7, USB 3.0*), debido a que han incluido 5 conectores extra. Es compatible con los estándares anteriores. Se espera que los productos fabricados con esta tecnología lleguen al consumidor entre 2009 y 2010.

2.2.4. Características de la transmisión

Las señales del USB se transmiten por un cable de par trenzado con impedancia de 90 Ω ± 15%, cuyos hilos se denominan D+ y D-. Estos, colectivamente, utilizan señalización diferencial en half-dúplex para combatir los efectos del ruido electromagnético en enlaces largos. D+ y D- suelen operar en conjunto y no son conexiones simples.

Los niveles de transmisión de la señal,en las versiones 1.0 y 1.1, varían de 0 a 0'3 V para bajos (ceros) y de 2'8 a 3'6 V para altos (unos), y en ±400 mV en alta velocidad para la versión 2.0.

En las primeras versiones, los alambres de los cables no están conectados a masa, pero en el modo de alta velocidad se tiene una terminación de 45 Ω a masa o un diferencial de 90 Ω para acoplar la impedancia del cable.

Este puerto sólo admite la conexión de dispositivos de bajo consumo, es decir, que tengan un consumo máximo de 100 mA por cada puerto; sin embargo, en caso de que

estuviese conectado un dispositivo que permite 4 puertos por cada salida USB (extensiones de máximo 4 puertos), entonces la energía del USB se asignará en unidades de 100 mA hasta un máximo de 500 mA por puerto.

Pin	Nombre	Color del cable	Descripción
1	VCC	Rojo	+5v
2	D−	Blanco	Data −
3	D+	Verde	Data +
4	GND	Negro	Tierra

Tabla 1. Descripción de los pines de la interfaz USB

2.2.5. Propiedades

Las propiedades del USB son las que se detallan a continuación:

- Conexión más sencilla – Sólo existe un tipo de cable (USA A-B) con conectores distintos en cada extremo, de manera que es imposible conectarlo erróneamente.
- Plug and Play – Cuando se conecta un dispositivo (memoria flash, impresora, cámara fotográfica, o escáner) a través de la interfase USB, no es necesario apagar el equipo, ni hacer una búsqueda manual, ya que el sistema automáticamente reconoce el dispositivo conectado e instala los controladores adecuados.
- Hot Pluggable – El usuario podrá conectar y desconectar los dispositivos USB las veces que quiera sin necesidad de apagar y encender el equipo.
- Mayor Rendimiento – La gran ventaja de usar el puerto USB es la velocidad de transferencia de los datos desde el ordenador al dispositivo de hasta 12 Mbps. Esto lo hace mucho más rápido que los puertos serie y paralelo.
- Soporte Multiplataforma – Responde a todas las necesidades de los usuarios con el mismo hardware para todas las plataformas.
- Múltiples Dispositivos Conectados de Manera Simultanea – La tecnología USB permite conectar hasta 127 dispositivos a un equipo a través de concentradores.

Todas estas ventajas propician que los fabricantes de equipos incorporen puertos de conexión USB en todas las unidades que producen.

2.2.6. Conectores y compatibilidad

El estándar USB especifica tolerancias para impedancia de los conectores, intentando minimizar la incompatibilidad entre los conectores fabricados por distintas compañías, objetivo que ya se ha logrado. El estándar USB, a diferencia de otros estándares también define tamaños para el área alrededor del conector de un dispositivo, para evitar el bloqueo de un puerto adyacente por el dispositivo en cuestión.

Las especificaciones definen dos tipos de conectores para conectar dispositivos al servidor: tipo A y tipo B. Sin embargo, la capa mecánica es distinta en algunos conectores aunque en todos se mantienen las señales y protocolos característicos del USB. Por ejemplo, el IBM UltraPort es un conector USB privado localizado en la parte

superior del LCD de los ordenadores portátiles de IBM. Otros fabricantes de artículos pequeños han desarrollado también sus medios de conexión pequeños, y ha aparecido una gran variedad de ellos, algunos de baja calidad (*véase figura 1*).

Existe además, una extensión del USB llamada "USB-On-The-Go" (sobre la marcha) que permite a un puerto actuar como servidor o como dispositivo dependiendo qué lado del cable esté conectado al aparato. Incluso después de que el cable está conectado y las unidades se están comunicando, las 2 unidades pueden "cambiar de papel" bajo el control de un programa.

Esta facilidad está específicamente diseñada para dispositivos del tipo PDA, donde el enlace USB podría conectarse a un PC como un dispositivo, y conectarse como servidor a un teclado o ratón.

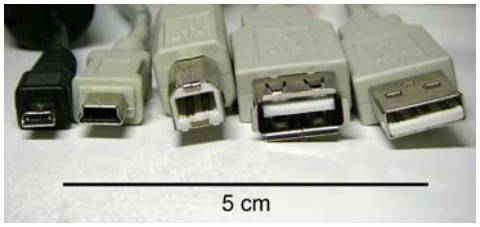


Figura 1. Distintos tipos de conectores USB (de izquierda a derecha): micro USB macho, mini USB tipo B macho, Tipo B macho, Tipo A hembra, Tipo A macho

2.2.7. USB 3.0

Presentado en el año 2008. Aunque está listo para su uso, es probable que pase entre uno o dos años, para ser incluido en dispositivos de uso masivo, lo que sitúa la aparición de productos con esta nueva especificación a partir del año 2009 o 2010.

La principal novedad técnica del puerto USB 3.0. será la inclusión de fibra óptica, lo cual eleva a 4.8 Gigabits/s la capacidad de transferencia que en la actualidad es de 480 Mb/s. Se mantendrá el cableado interno de cobre para asegurarse la compatibilidad con las tecnologías USB 1.0 y 2.0.

Si en USB 2.0 el cable dispone de cuatro líneas, un par para datos, una de corriente y una de toma de tierra, en USB 3.0 se añade cinco líneas. Dos de ellas se usarán para el envío de información y otras dos para la recepción, de forma que se permite el tráfico bidireccional, en ambos sentidos al mismo tiempo. El aumento del número de líneas permite incrementar la velocidad de transmisión desde los 480 Mb/s hasta los 4,8 Gb/s. De aquí se deriva el nombre que también recibe esta especificación: USB *Superspeed*.

La cantidad de energía que transporta un cable USB 1.x y 2.0 resulta insuficiente en muchas ocasiones para recargar algunos dispositivos, especialmente si utilizamos concentradores donde hay conectados varios de ellos. En USB 3.0, se aumenta la intensidad de la corriente de 100 miliamperios a 900 miliamperios, con lo que pueden ser cargados más dispositivos o hacerlo más rápido. Este aumento de la intensidad podría traer consigo un menor rendimiento energético.

Pensando en esta idea, USB 3.0 utiliza un nuevo protocolo basado en interrupciones, al contrario que el anterior que se basaba en consultar a los dispositivos periódicamente.

El aumento de líneas en USB 3.0 provoca que el cable sea más grueso, un inconveniente importante. Si hasta ahora los cables eran flexibles, con el nuevo estándar estos tienen un grueso similar a los cables que se usan en redes Ethernet, siendo por tanto más rígidos. Al igual que pasa entre USB 1.1 y USB 2.0 la compatibilidad está garantizada entre USB 2.0 y USB 3.0, gracias al uso de conectores similares, cuyos contactos adicionales se sitúan en paralelo, de forma que no afectan en caso de usar algún puerto que no sea del mismo tipo.

2.3. Memorias Flash USB

Una memoria USB (Universal Serial Bus, en inglés pendrive o USB flash drive) es un pequeño dispositivo de almacenamiento que utiliza memoria flash para guardar la información. Es un dispositivo cuyos elementos básicos de memoria son transistores y por tanto se denomina también memoria de semiconductor. Su protocolo de comunicación con el exterior es lo que las ha hecho tan útiles y ha facilitado enormemente su difusión. Estas memorias son resistentes a los rasguños externos y al polvo que han afectado a las formas previas de almacenamiento portátil, como los disquetes, CDs y los DVDs. [20]

Estas memorias se han convertido en el sistema de almacenamiento y transporte personal de datos más utilizado, desplazando a los tradicionales disquetes, y a los CDs. Se pueden encontrar en el mercado fácilmente memorias de 1, 2, 4, 8, 16, 32 GB o más (esto supone, como mínimo, el equivalente a unos 1000 disquetes). Su gran popularidad le ha supuesto infinidad de denominaciones relacionadas con su pequeño tamaño y las diversas formas de presentación, sin que ninguna haya podido destacar entre todas ellas: pincho, lápiz, mechero, llavero, llave o las de los embalajes originales en inglés *pendrive*, *flash drive* o *memory stick*. El calificativo USB o el propio contexto permite identificar fácilmente el dispositivo informático al que se refieren.

Los sistemas operativos actuales pueden leer y escribir en las memorias sin más que conectarlas a una interfaz USB del equipo, recibiendo la energía de alimentación a través del propio conector USB.

2.3.1. Historia

Las unidades flash USB fueron inventadas en 1995 por IBM como un reemplazo de las unidades de disquete para su línea de productos ThinkPad. Aunque fue un invento de IBM, ésta no lo patentó. IBM contrató más tarde a M-Systems para desarrollarlo y fabricarlo en forma no exclusiva. M-Systems mantiene la patente de este dispositivo.

Las primeras unidades flash fueron fabricadas por M-Systems bajo la marca "Disgo" en tamaños de 8 MB, 16 MB, 32 MB y 64 MB. Estos fueron promocionados como los "verdaderos reemplazos del disquete", y su diseño continuó hasta los 256 MB. Los fabricantes asiáticos pronto fabricaron sus propias unidades, más baratas que las de la serie Disgo.

Las modernas unidades flash (2009) poseen conectividad USB 2.0 y almacenan hasta 64 GB de memoria.

2.3.2. Componentes de las memorias flash USB

Las partes principales de una memoria USB son las mostradas en la figura 2.

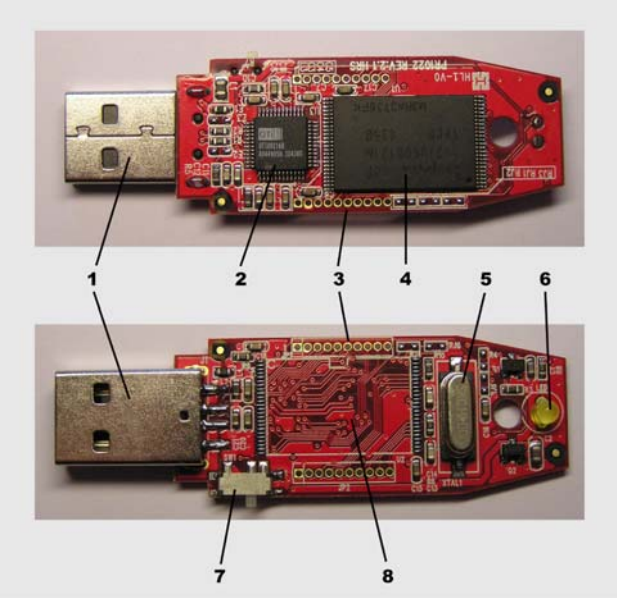


Figura 2. Partes de una memoria flash USB

- Un conector USB macho tipo A (1): Provee la interfaz física con la computadora.
- Controlador USB de almacenamiento masivo (2): Implementa el controlador USB y provee la interfaz homogénea y lineal para dispositivos USB seriales orientados a bloques, mientras oculta la complejidad de la orientación a bloques, eliminación de bloques y balance de desgaste. Este controlador posee un pequeño microprocesador RISC y un pequeño número de circuitos de memoria RAM y ROM.
- Circuito de memoria Flash NAND (4): Almacena los datos.
- Oscilador de cristal (5): Produce la señal de reloj principal del dispositivo a 12 MHz y controla la salida de datos a través de un bucle de fase cerrado (phase-locked loop).

Adicionalmente también puede contar con:

- Puentes y Puntos de prueba (3): Utilizados en pruebas durante la fabricación de la unidad o para la carga de código dentro del procesador.
- LEDs (6): Indican la transferencia de datos entre el dispositivo y la computadora mediante secuencias de apagado y encendido. Esto permite chequear que el dispositivo está funcionando correctamente.
- Interruptor para protección de escritura (7): Utilizado para proteger los datos de operaciones de escritura o borrado.
- Espacio Libre (8): Se dispone de un espacio para incluir un segundo circuito de memoria. Esto le permite a los fabricantes utilizar el mismo circuito impreso para dispositivos de distintos tamaños y responder así a las necesidades del mercado.

- Tapa del conector USB: Reduce el riesgo de daños y mejora la apariencia del dispositivo. Algunas unidades no presentan una tapa pero disponen de una conexión USB retráctil. Otros dispositivos poseen una tapa giratoria que no se separa nunca del dispositivo y evita el riesgo de perderla.
- Ayuda para el transporte: En muchos casos, la tapa contiene una abertura adecuada para una cadena o collar, sin embargo este diseño aumenta el riesgo de perder el dispositivo. Por esta razón muchos otros tiene dicha abertura en el cuerpo del dispositivo y no en la tapa, la desventaja de este diseño está en que la cadena o collar queda unida al dispositivo mientras está conectado. Muchos diseños traen la abertura en ambos lugares.

2.3.3. Utilidades

Las memorias USB son pequeñas y ligeras. Son comunes entre personas que transportan datos entre su casa y el lugar de trabajo. Teóricamente pueden retener los datos durante unos 10 años y escribirse un millón de veces.

Aunque inicialmente fueron concebidas para guardar datos y documentos, es habitual encontrar en las memorias USB programas o archivos de cualquier otro tipo.

La disponibilidad de memorias USB de bajo precio ha provocado que sean muy utilizadas con objetivos promocionales o de marketing, especialmente en ámbitos relacionados con la industria de la computación. A menudo se distribuyen de forma gratuita, se venden por debajo del precio de coste o se incluyen como obsequio al adquirir otro producto.

Las memorias USB pueden ser configuradas con la función de autoarranque (*autorun*) para Microsoft Windows, con lo que al insertar el dispositivo arranca de forma automática un archivo específico. Para activar la función *autorun* es necesario guardar un archivo llamado *autorun.inf* con el *script* apropiado en el directorio raíz del dispositivo. En ocasiones esta funcionalidad se encuentra deshabilitada para dificultar la propagación de virus y virus tipo Caballo de Troya que se aprovechan de este sistema de arranque.

Otra utilidad de estas memorias es que, si la *BIOS* del equipo lo admite, pueden arrancar un sistema operativo sin necesidad de disco duro. El arranque desde memoria USB está muy extendido en ordenadores nuevos y es más rápido que con un lector de *DVD-ROM*. Se pueden encontrar distribuciones de *GNU/Linux* que están contenidas completamente en una memoria USB y pueden arrancar desde ella.

Las memorias USB de gran capacidad, al igual que los discos duros o grabadores de *CD/DVD* son un medio fácil para realizar una copia de seguridad, por ejemplo.

Además, en la actualidad, existen equipos de audio con entradas USB a los cuales podemos conectar dichas memorias y reproducir la música contenida en las mismas.

2.3.4. Fortalezas y debilidades

A pesar de su bajo coste y garantía, hay que tener muy presente que estos dispositivos de almacenamiento pueden dejar de funcionar repentinamente por accidentes diversos: variaciones de voltaje mientras están conectadas, por dejarlas caer de una altura superior a un metro, por su uso prolongado durante varios años especialmente en dispositivos antiguos.

Las unidades flash son inmunes a rayaduras y al polvo que afecta a las formas previas de almacenamiento portátiles como discos compactos y disquetes. Su tecnología de estado sólido duradero significa que en muchos casos puede sobrevivir a accidentes ocasionales, como golpes, caídas, pisadas, pasadas por la lavadora o salpicaduras de líquidos. Esto lo hace ideal para el transporte personal de datos, archivos de trabajo o datos personales a los que se quiere acceder en múltiples lugares.

La casi omnipresencia de soporte USB en equipos modernos significa que un dispositivo funcionará en casi todas partes.

Las unidades flash son una forma relativamente densa de almacenamiento, hasta el dispositivo más barato almacenará lo que decenas de disquetes, y por un precio moderado alcanza a los CD en tamaño o los superan. Históricamente, el tamaño de estas unidades ha ido variando de varios megabytes hasta unos pocos gigabytes.

En el año 2003 las unidades con interfaz USB 1.0/1.1, funcionaban a velocidades de unos 1.5 Mbit/s o 12 Mbit/s. En 2004 aparecieron los primeros dispositivos con interfaz USB 2.0, que pueden alcanzar hasta los 480 Mbit/s. En este caso la limitación está en el ancho de banda del dispositivo de memoria interno, por lo que la velocidad de lectura se reduce a unos 100 Mbit/s, y con operaciones de escritura aún más lentas. En condiciones óptimas, un dispositivo USB puede retener información durante unos 10 años.

Las memorias flash implementan el estándar USB *mass storage device class* (clase de dispositivos de almacenamiento masivo USB). Esto significa que la mayoría de los sistemas operativos modernos pueden leer o escribir en dichas unidades sin drivers adicionales. En lugar de exponer los complejos detalles técnicos subyacentes, los dispositivos flash exportan una unidad lógica de datos estructurada en bloques al sistema operativo anfitrión. El sistema operativo puede usar el sistema de archivos o el esquema de direccionamiento de bloques que desee.

Las memorias flash pueden soportar un número finito de ciclos de lectura/escritura antes de fallar, Con un uso normal, el rango medio es de alrededor de varios millones de ciclos. Sin embargo las operaciones de escrituras serán cada vez más lentas a medida que la unidad envejezca. Esto debe tenerse en consideración cuando usamos un dispositivo flash para ejecutar desde ellas aplicaciones de software o un sistema operativo. Para manejar esto (además de las limitaciones de espacio en las unidades comunes), algunos desarrolladores han lanzado versiones de sistemas operativos (como Linux) o aplicaciones comunes (como Mozilla Firefox) diseñadas especialmente para ser ejecutadas desde unidades flash. Esto se logra reduciendo el tamaño de los archivos de intercambio y almacenándolos en memoria RAM.

2.4. Criptografía

La finalidad de la criptografía es la de garantizar el secreto en la comunicación entre dos entidades (personas, organizaciones, etc.) y asegurar que la información que se envía es auténtica en el doble sentido de que el remitente sea realmente quien dice ser y que el contenido del mensaje enviado no haya sido modificado en su tránsito. [6]

Los términos más importantes que convienen saber son los siguientes:

- Texto en claro – La información original que debe protegerse.
- Texto cifrado o criptograma – Es texto ilegible, que contiene de forma oculta el texto claro.
- Cifrado – Es el proceso mediante el cual se convierte el texto plano en texto cifrado o criptograma. Para realizar este proceso se utiliza un algoritmo matemático que se conoce como algoritmo de cifrado. Los algoritmos de cifrado se basan en la existencia de una clave.
- Descifrado – Es el proceso inverso que recupera el texto plano a partir del criptograma y la clave.
- Clave – Es secreta y utiliza el algoritmo de cifrado para ocultar información.
- Criptosistema o sistema criptográfico – Se define como los fundamentos y procedimientos de operación que participan en el cifrado y descifrado de un mensaje. Todo sistema criptográfico consta de cinco componentes:
  - M – Es el mensaje o texto plano que va a ser cifrado.
  - C – Es el texto cifrado o criptograma que obtenemos a partir de M.
  - K – Es el conjunto de claves a utilizar.
    - K_C – Clave de cifrado.
    - K_D – Calve de descifrado.
  - E – Es el método o algoritmo de cifrado. $E = \{E_k / M \rightarrow C, \forall k \in K\}$
  - D – Es el método o algoritmo de descifrado. $D = \{D_k / C \rightarrow M, \forall k \in K\}$

En la tabla 2 se muestra resumen simplificado de los sistemas criptográficos:

<table>
<tr><td rowspan="3">Criptografía clásica o convencional</td><td>Sustitución Simple</td><td colspan="3">Cesar</td></tr>
<tr><td>Polialfabético</td><td colspan="3">Vigenere</td></tr>
<tr><td>Trasposición</td><td colspan="3">Escitala Espartana</td></tr>
<tr><td rowspan="8">Criptografía moderna</td><td rowspan="2">Cifrado en flujo</td><td colspan="3">A5</td></tr>
<tr><td colspan="3">RC4</td></tr>
<tr><td rowspan="6">Cifrado en bloque</td><td rowspan="5">Clave privada</td><td colspan="2">DES</td></tr>
<tr><td colspan="2">Triple DES</td></tr>
<tr><td colspan="2">IDEA</td></tr>
<tr><td colspan="2">AES</td></tr>
<tr><td colspan="2">RC5</td></tr>
<tr><td rowspan="3">Clave pública</td><td rowspan="2">Exponenciación</td><td>RSA</td></tr>
<tr><td>ElGamal</td></tr>
<tr><td>Suma / Producto</td><td>Curvas Elípticas</td></tr>
</table>

Tabla 2. Resumen de los sistemas criptográficos

2.4.1. Algoritmo de cifrado DES

DES (Data Encryption Standard) es un algoritmo de cifrado, es decir, un método para cifrar información, escogido como estándar en 1976 en los Estados Unidos, y propagado posteriormente por todo el mundo. [8]

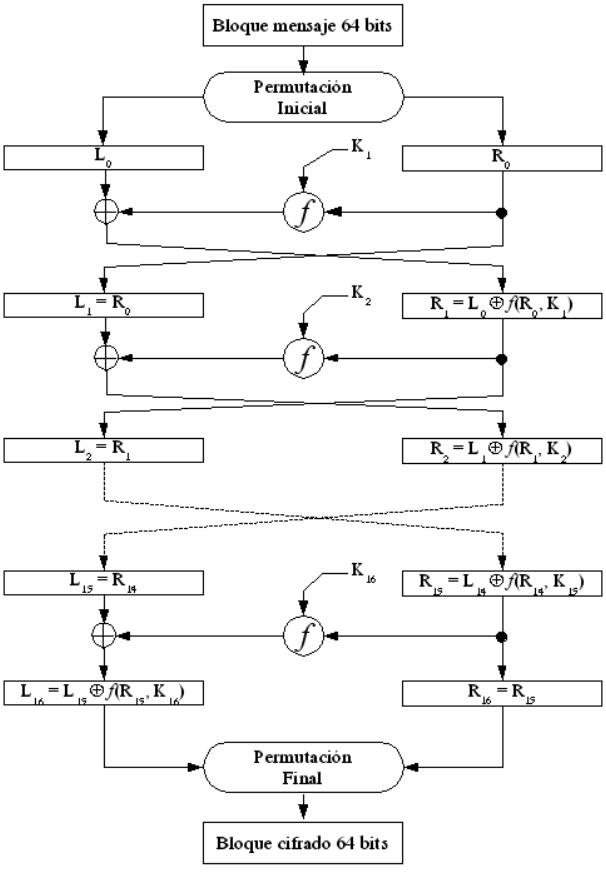
DES es el algoritmo prototipo del cifrado por bloques, es decir, un algoritmo que toma un texto en claro de una longitud fija de bits y lo transforma mediante una serie de operaciones matemáticas y lógicas, en otro texto cifrado de la misma longitud.

En el caso de DES el tamaño del bloque es de 64 bits. Además utiliza una clave criptográfica para modificar la transformación, de modo que el descifrado sólo se realiza correctamente en el caso de conocerse la clave concreta utilizada en el cifrado. Dicha clave es de 64 bits, aunque en realidad, sólo 56 de ellos son empleados por el algoritmo. Los ocho bits restantes se utilizan únicamente para comprobar la paridad, y después son descartados.

Al igual que otros cifrados de bloque, DES debe ser utilizado en el modo de operación de cifrado de bloque si se aplica a un mensaje mayor de 64 bits. Los modos de operación de cifrado de bloque son los que se especifican a continuación:

- *Electronic Code Book* (ECB)
- *Cipher-Block Chaining* (CBC)
- *Cipher Feedback* (CFB)
- *Output Feedback* (OFB)
- *Counter* (CTR)

La estructura básica del algoritmo consta de 16 fases idénticas de proceso, denominadas rondas. También hay una permutación inicial y final, que son funciones inversas entre sí. Antes de las rondas, el bloque es dividido en dos mitades de 32 bits y procesadas alternativamente. El entrecruzamiento que se realiza se conoce como función o esquema Feistel.



La función de Feistel hace que el cifrado y el descifrado sean procesos muy similares, con la única diferencia de que las subclaves se aplican en orden inverso cuando desciframos. Esto simplifica la implementación, en especial sobre hardware.

La función F mezcla la mitad del bloque con parte de la clave. La salida de la función F se combina entonces con la otra mitad del bloque, y los bloques son intercambiados antes de la siguiente ronda. Tras la última ronda, las mitades no se intercambian gracias a lo cual se consigue que el cifrado y descifrado sean parecidos.

La función de Feistel consta de cuatro pasos:

1. **Expansión** – La mitad del bloque con 32 bits se expande a 48 bits, duplicando algunos de los bits.
2. **Mezcla** – El resultado se combina con una subclave utilizando una operación XOR. A parte de la clave inicial se obtienen 16 subclaves, una para cada ronda.
3. **Sustitución** – Tras mezclarlo con la subclave, el bloque es dividido en ocho trozos de 6 bits. Cada una de las ocho cajas S reemplaza sus 6 bits de entrada con 4 bits de salida, de acuerdo con una trasformación no lineal, especificada por una tabla de búsqueda. Las cajas S constituyen el núcleo de la seguridad de DES.
4. **Permutación** – Finalmente, las 32 salidas de las cajas S se reordenan de acuerdo a una permutación fija P.

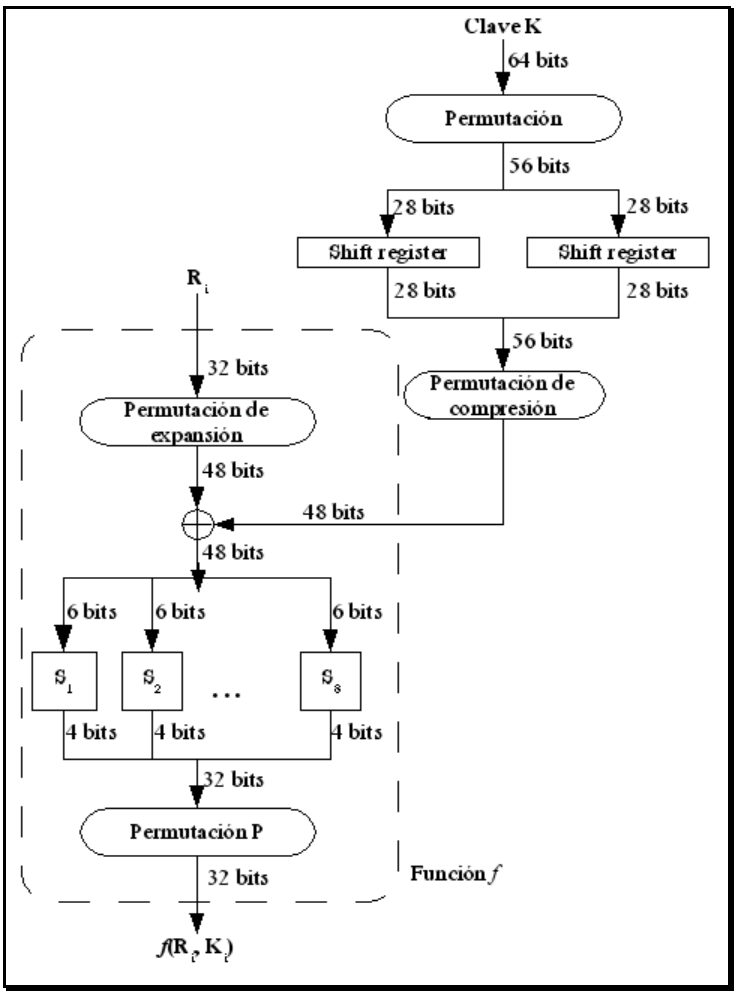


Figura 3. Pasos de la función de Feistel para el algoritmo DES

3. Entorno de trabajo

En este apartado se describe el entorno de desarrollo de CALUS, tanto el hardware con el que se ha contado como las herramientas software que se han utilizado.

3.1. Infraestructura de desarrollo

Tanto para el desarrollo del sistema CALUS, como para su explotación ha sido necesario contar con varios equipos informáticos. Algunos de ellos ya existían anteriormente, pero ha habido que incorporar algunos nuevos para hacer que el desarrollo y la puesta en funcionamiento de CALUS fueran posibles.

En este apartado se describirán todos los equipos hardware y las herramientas software que han participado en el proceso de desarrollo de este proyecto. Las decisiones acerca del software elegido se describirán mas adelante (*véase sección 3.2, Herramientas software*).

Se puede observar el entorno de trabajo del sistema CALUS en la figura 4:

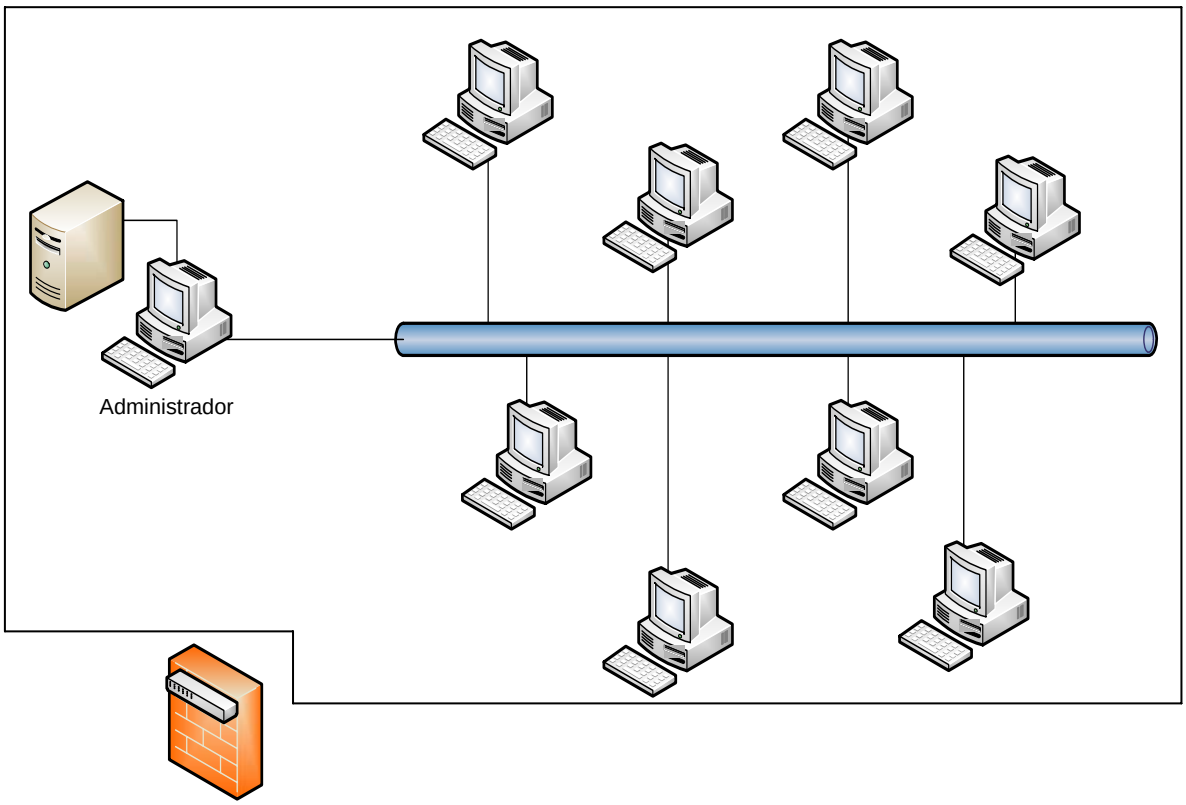


Figura 4. Infraestructura de un laboratorio o aula informática

A continuación se describirá cada uno de los elementos pertenecientes al entorno de trabajo:

- Red de los laboratorios – Es una red local de ordenadores que interconecta todos los equipos del aula entre sí, y además está conectada a Internet.

- Puestos de aulas – Es el conjunto de todos los puestos que pertenecen a las Aulas Informáticas Generales de la Universidad. Desde cada uno de ellos, un usuario podrá iniciar una sesión validándose a través del sistema CALUS, podrá acceder a Internet y utilizar la aplicación de gestión de CALUS. Principales características:
  - Nº equipos: Más de 1000.
  - Procesador: Pentium III o superior.
  - Sistema Operativo: Microsoft Windows NT 4.0 Workstation, Windows XP.
  - Cliente de validación CALUS.
  - Software de docencia.
  - Interfaz de red: FastEthernet 100Mbps.
- Dispositivo de acceso en las puertas – Es una interfaz que facilita el acceso al personal de la universidad (estudiantes, PAS, PDI).
- Equipos de desarrollo – Son los equipos que se encuentran en el laboratorio de investigación del grupo de Diseño Microelectrónico y Aplicaciones del departamento de Tecnología Electrónica. Las principales características son:
  - Nº equipos: 20.
  - Procesador: Intel Pentium III 650 MHz.
  - Sistema Operativo: Windows XP o Vista
  - Cliente de validación CALUS.
  - Eclipse.
  - Compilador de C: GCC.
  - VMWare.
  - Microsoft Project 2007.

3.2. Herramientas software

En las siguientes secciones se describirán las herramientas de software utilizadas en el proceso de desarrollo de este proyecto.

3.2.1. Eclipse

Eclipse es un entorno de programación en el que se combinan la programación orientada a objetos (Java) y un sistema de desarrollo diseñado para crear aplicaciones tanto de consola como gráficas para Windows o Linux (Software Development Kit: SDK).

Aunque las aplicaciones Java son sencillas de utilizar e integrar en cualquier sistema operativo, el desarrollo de las mismas no es fácil. Para facilitar el desarrollo, Eclipse incluye varias herramientas que lo convierten en un potente generador de programas, así como un sistema de gestión de las bibliotecas de Java que permiten crear de una forma intuitiva las aplicaciones.

Además, incluye plugins de todo tipo, que permiten desde instalar un servidor *SVN* o diseño de interfaces graficas, hasta la programación en otros lenguajes incluyendo los orientados a componentes.

Eclipse incluye, como características más destacadas:

- Un entorno de desarrollo integrado (editor, compilador, depurador, analizador, administrador de proyectos, etc).
- Bibliotecas de funciones para desarrollo de los programas de la aplicación: *Application Procedural Interface* (API) completos de Java y de otros lenguajes instalados mediante *plugin*.
- Resaltado de sintaxis.
- Compilación en tiempo real.
- Interfaz para múltiples documentos que permite crear una aplicación con una ventana de aplicación y múltiples ventanas de documento.
- Un sistema de ayuda en línea que incluye los manuales.
- Soporte para dibujar el interfaz gráfico del usuario.
- Asistentes (wizards): para creación de proyectos, clases, tests, etc.

Cuando se combinan estas características, se dispone de un potente sistema de desarrollo que permite diseñar eficientemente aplicaciones sofisticadas.

3.2.2. uClinux SDK

uClinux SDK es un entorno de desarrollo que trabaja bajo el sistema operativo Debian de Linux. En este proyecto se ha utilizado para el desarrollo de la aplicación que funciona dentro de la tarjeta LCP2468.

Incluye todas las bibliotecas de gestión y control de cada uno de los módulos que integran las tarjetas LPC24xx. Esta basado en un compilador de lenguaje C cruzado, para que las aplicaciones sean compatibles con el entorno de ejecución de las tarjetas LPC24xx (*véase sección 6.1, Configuración del equipo hardware*).

Además el sistema operativo Debian proporciona el entorno de programación basado en editores de texto con resaltado de sintaxis.

uClinux incluye, como características más destacadas:

- Un entorno de desarrollo no integrado pero de fácil utilización (editor, compilador y depurador).
- Bibliotecas de funciones para desarrollo de los programas de la aplicación: *Application Procedural Interface* (API) completos de C.
- Resaltado de sintaxis.
- Compilación cruzada para las tarjetas LCP24xx.

Cuando se combinan estas características, se dispone de un potente sistema de desarrollo que permite diseñar eficientemente aplicaciones sofisticadas integrables en las tarjetas LPC24xx.

4. Descripción del sistema.

El nuevo marco en el que se sitúa la educación universitaria europea (Espacio Europeo de Educación Superior, EEES), hace del trabajo personal del alumno una parte crucial para su formación. En este concepto de trabajo personal no sólo se incluye el estudio de los temas o resolución de problemas; en las titulaciones técnicas el alumno debe adquirir experiencia en el manejo de aplicaciones software y sistemas de desarrollo hardware/software que no son económicamente accesibles para los alumnos.

Sin embargo, las licencias del campus y la capacidad para compartir recursos, hacen posible el uso de las últimas novedades tecnológicas en la universidad pública española.

En el nuevo EEES, el alumno debe enfrentarse de forma individual a la adquisición de experiencia en el uso de estos sistemas. Es necesario por tanto dotar a la Escuela Politécnica Superior de la UC3M de mecanismos de acceso controlado a los laboratorios, junto a un sistema de seguimiento del trabajo realizado, que pueda ayudar al profesor en la tarea de evaluación del alumno.

Con este sistema se genera la plataforma básica para facilitar la inclusión de actividades de autoevaluación del alumno o evaluación remota por parte del profesor. Cuestionarios on-line, entrega de trabajos, auto-corrección, foros de discusión, etc.

En este PFC (Proyecto Fin de Carrera) se plantea el diseño y desarrollo de un sistema para gestionar de una manera eficaz, el acceso a zonas delicadas, como son los laboratorios y aulas informáticas de la universidad permitiendo el acceso controlado a personal autorizado, así como registrar cada uno de los accesos a dichas zonas.

Este sistema ofrece un método seguro de control mediante memorias flash con interfaz USB, utilizando cierres sólidos y mecánicamente fiables. Todo el control se realiza bajo la supervisión de un conjunto de equipos hardware que utilizan software de procesamiento de datos y fácil manejo. El software registra todas las actividades y proporcionar informes concisos, y situaciones de no-aceptación de los usuarios.

El concepto de llave USB hace referencia a una memoria flash de almacenamiento masivo con interfaz USB (*véase sección 2.3, Memorias Flash USB*). En este trabajo, dicha llave almacena en su interior un fichero cifrado con los datos del usuario, que le permite el acceso a ciertas zonas controladas dentro de la universidad. Además, cada uno de los usuarios tiene acceso a una serie de herramientas relacionadas con las asignaturas en las que está matriculado.

Tan sólo con estas llaves, un usuario puede obtener acceso a un laboratorio o aula informática, e iniciar su sesión personal en cualquiera de los equipos informáticos. Así cada usuario puede ser identificado de forma única por el sistema, lo que facilita, además del control de acceso, la generación de estadísticas de uso de los equipos, que posteriormente serán utilizadas por el administrador.

El sistema se subdivide en dos secciones:

- **Subsistema de control de acceso al área restringida** – Es predominantemente hardware. Este sistema esta formado por un microprocesador integrado en una tarjeta con distintos controladores que gestionan los periféricos integrados en la misma. Irá instalado dentro de una caja protectora y ubicado en la pared, a un lado de la puerta que da acceso al área restringida. El dispositivo, por tanto controlara el acceso de personal autorizado.
- **Subsistema de control de acceso a los equipos informativos –** Es predominantemente software. Los equipos controlados serán ordenadores (PC's) con el sistema operativo Window XP (aunque puede ampliarse para otro tipo de equipos) y con entrada USB. Con este sistema se permitirá el acceso a los equipos sólo al personal autorizado y se llevará un control de su actividad.

4.1 Especificación del subsistema de control de acceso a un área restringida (Subsistema Hardware)

El principal objetivo del sistema es controlar el acceso a los laboratorios, laboratorios docentes y aulas informáticas. Esto ha de ser gestionado por una interfaz hardware que cuente con las siguientes características:

- Disponer de un microprocesador de alta velocidad y bajo consumo, para evitar utilizar grandes fuentes de alimentación, ya que el módulo irá integrado en la pared. Habrá tantos dispositivos de este tipo como laboratorios se incluyan en el sistema.
- El resto del hardware, que incluye la PCB con los periféricos, también ha de ser de bajo consumo por la restricción en las fuentes de alimentación anteriormente comentada.
- Disponer de una interfaz USB con funcionamiento en modo host, y de la electrónica necesaria para realizar la configuración y la comunicación con el microprocesador.
- Tener la capacidad de conexión con una pantalla LCD en color y táctil, que permitirá a los usuarios interactuar con el sistema para autenticarse.
- Contar con una interfaz MMC/SD para poder introducir una tarjeta de memoria con el propósito de almacenar los datos de acceso y que puedan ser recuperados con posterioridad por el administrador del sistema de control de acceso.
- Integrar una interfaz de red y otra USB para la comunicación directa entre un PC y la tarjeta hardware, que además deberá permitir carga y descarga de aplicaciones y datos.
- Implementar la electrónica necesaria para la apertura y cierre de puertas de forma automática.

4.2. Especificación del subsistema de control de acceso a los equipos (Subsistema Software)

La aplicación de control de acceso a los equipos de los laboratorios está dividida en dos partes. Por un lado se plantea desarrollar el software referente a las máquinas que pueden utilizar los usuarios, y por otro realizar un software de menor tamaño, que irá instalado en el equipo del administrador, y que será el encargado, principalmente, de gestionar los ficheros de estadísticas de uso de los equipos por parte de los usuarios (alumnos, PDI, PAS).

El principal objetivo del sistema de control de acceso, es impedir que un usuario no autorizado acceda a los equipos que haya en un laboratorio. Partiendo de esa idea se ha decidido desarrollar una aplicación basada en Java que implemente una interfaz grafica bloqueante, bajo la cual se gestione dicho control de acceso, y que además sólo lleve un control del trabajo de las herramientas CAD relacionadas con sus asignaturas. Esta aplicación software tiene las siguientes características:

- Impedir el acceso a los equipos mientras el usuario no se haya autenticado contra el sistema. El bloqueo se hará mediante una interfaz grafica que a su vez indicará al usuario las instrucciones que debe seguir para autenticarse.
- Contar con una interfaz USB y con un sistema de control de dichos puertos en los equipos de los usuarios, para detectar los eventos relacionados con la introducción de las llaves USB en dichos equipos.
- Implementar un sistema de control de la actividad del usuario, relacionada con aspectos académicos (sus asignaturas), y administrar un sistema de estadísticas de uso de determinados programas. Estos datos se tienen que enviar al administrador, y serán utilizados posteriormente como ayuda a la evaluación de los alumnos.
- Establecer el sistema básico para incluir actividades de evaluación remota, autoevaluación, entrega de cuestionarios y/o prácticas, foros de debate *on-line*, teleaprendizaje, etc.
- Debe existir un sistema de gestión y control en el equipo del administrador que reciba los datos de uso de los equipos por parte de los usuarios. Para ello se implementará un sistema de comunicaciones entre los equipos de los usuarios y el sistema del administrador, con el objetivo de que realicen copias de las estadísticas tanto en la memoria flash USB de los alumnos (llaves USB) como en un equipo de la universidad (PC del administrador).
- Implementación de una interfaz sencilla y de fácil acceso que permita a los usuarios crear los ficheros cifrados para autenticarse, que irán dentro de las memorias flash USB.

5. Diseño y desarrollo

5.1. Implementación de la aplicación en C para el subsistema hardware

En este apartado se detalla la implementación de la aplicación en C [7] [9] para llevar a cabo el desarrollo de las distintas funciones del sistema de control de acceso por hardware, así como el API[1] (Application Program Interface) que se ha utilizado.

5.1.1. Diseño de la aplicación

Este subsistema consiste en un dispositivo hardware que controla el acceso a laboratorios docentes y aulas informáticas (en adelante zonas restringidas), mediante dispositivos de apertura y cierre automáticos de la puerta de acceso.

El software de control se encarga de gestionar tanto el proceso de validación, como de almacenar las estadísticas de todos los accesos realizados durante un periodo de tiempo. Además maneja el sistema de apertura y cierre de la puerta activando o desactivando el dispositivo de control.

Todo el equipo, salvo la electrónica de la puerta de acceso, se concentra en una caja ubicada en la pared, al lado de la puerta. Desde el punto de vista del usuario, el sistema se compone de una pantalla LCD táctil QVGA en color de 3.2 pulgadas, y un puerto de entrada USB.

El diagrama de bloques que describe la composición es el que se muestra en la figura 5.

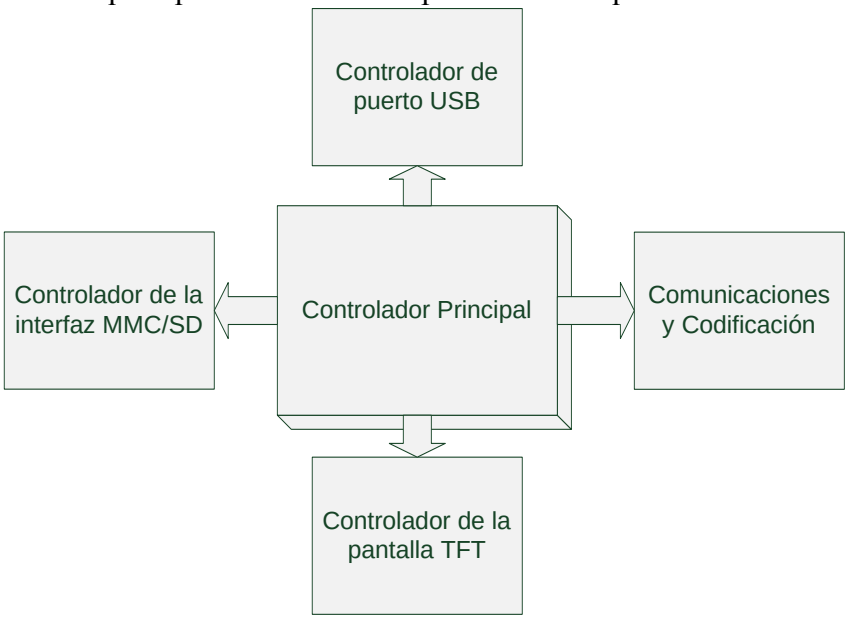


Figura 5. Diagrama de bloques del software de control de acceso hardware

[1] Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación. Cuando se intenta estandarizar una plataforma, se estipulan unos APIs comunes a los que deben ajustarse todos los desarrolladores de aplicaciones.

A continuación se presentan las distintas pantallas que se mostrarán para cada uno de los estados del software de control de la tarjeta.

En primer lugar, la figura 6 muestra la pantalla de bienvenida con unas breves instrucciones de lo que el usuario deba hacer para poder tener acceso al laboratorio o sala controlada.

Figura 6. Diagrama de bloques del software de control de acceso hardware

Una vez que el usuario haya introducido una memoria USB que contenga un fichero llave con sus datos, se presenta otra pantalla con un teclado virtual que facilitará al usuario la introducción de su contraseña de acceso. En caso de que se produzca un error en la lectura del fichero, o simplemente que el fichero llave no exista, se mostrará una ventana con un mensaje de error indicando dicho fallo.

Como se puede ver en la figura 7 izquierda, la ventana de introducción de clave mediante teclado virtual tiene una "X" en la esquina inferior derecha, que permite al usuario cancelar el proceso de autenticación en cualquier momento.

Figura 7. Teclado virtual (izda) y mensaje de error al manejar el fichero llave (dcha).

Si la contraseña introducida es correcta, entonces se mostrará una ventana indicando al usuario que todo ha ido bien y que por tanto tiene acceso a la sala. Mientras tanto el dispositivo abrirá la puesta automáticamente. En el caso de que la contraseña fuera incorrecta aparecerá otra ventana indicando en este caso que ha habido un fallo de acceso. En ambos casos se almacenará en un fichero de log cada uno de los accesos tanto correctos como incorrectos.

Figura 8. Mensaje permiso de acceso (izda) y mensaje de denegación de acceso (dcha).

A continuación se detalla el subsistema hardware utilizado para el desarrollo de la aplicación. Este subsistema está basado en la tarjeta LPC2468 OEM Board de Embedded Artists, que integra un microprocesador de tipo ARM [10].

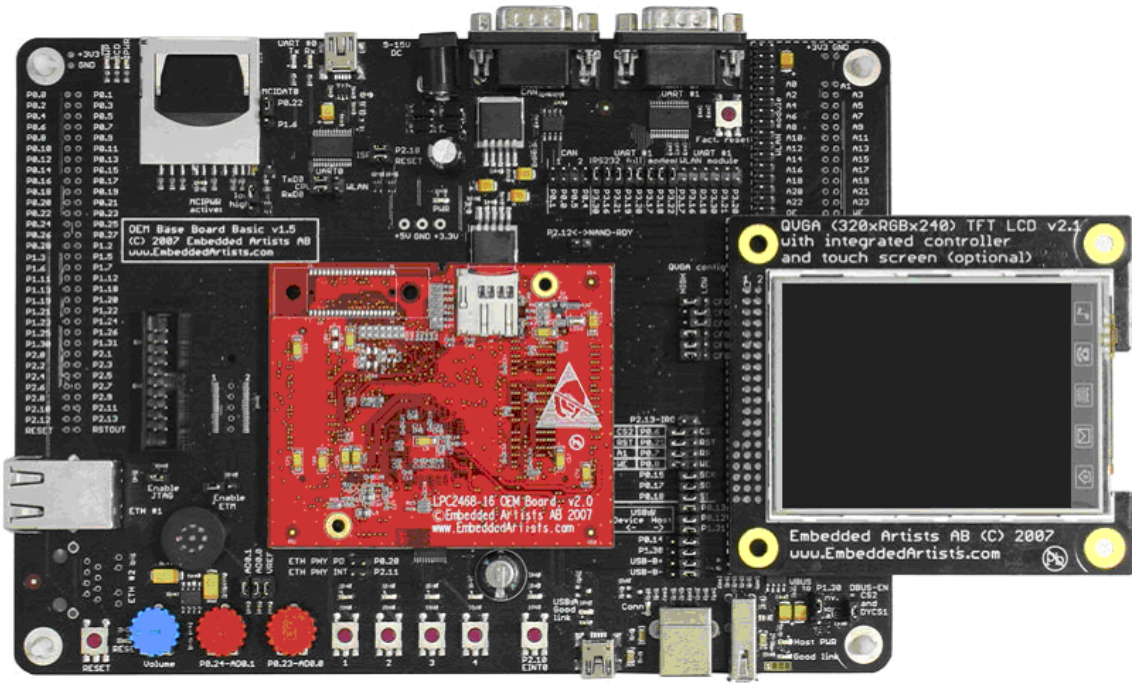


Figura 9. Tarjeta LPC2468 OEM Board

Este sistema se divide en tres partes:

- **LPC2468 OEM Board –** Esta tarjeta integra además del microprocesador ARM, distintos controladores (*véase especificación en la tabla 3*).
- **OEM Base Board –** Es una tarjeta entrenadora cuenta con numerosos periféricos y puertos de expansión (*véase especificación en la tabla 4*).
- **Pantalla LCD 3.2 inch QVGA Color–** Es un módulo a parte, que se conecta directamente a uno de los puertos de expansión de la tarjeta base. Existen dos modos de conexión con el bus de memoria de la tarjeta, en paralelo (8 o 16 bits) o en serie (*véase especificación en la tabla 5*).

Las características de cada una de estos componentes se describen a continuación:

LPC2468 OEM Board	
Características	**Descripción**
Procesador	Microcontrolador ARM7TDMI LPC2468 con encapsulado BGA
Programación sobre Memoria Flash	128 MB NAND FLASH + 4 MB NOR FLASH + 512 kB internos
Memoria de Datos	• 32 MB SDRAM + 96 KB internos • 16-bit data bus to SDRAM
Ethernet	Interfaz 100/10M Ethernet basada en Micrel KSZ8001L Ethernet PHY
Relojes/Cristales	• Cristal de 12.000 MHz para la CPU • Cristal de 32.768 kHz para el RTC (Real Time Clock)
Dimensiones	66 x 80 mm
Alimentación	• +3.3V (alimentación principal)
Conectores	• 2 conectores de 100 pines, 0.6mm pitch Hirose FX8C-100 para la OEM Board
Otros	• Chip ISP1301 de USB-OTG • 256 Kbit I2C E2PROM para almacenamiento de parámetros no volátiles • Bus de datos de 16-bit

Tabla 3. Características de la LPC2468 OEM Board

OEM Base Board	
Características	**Descripción**
Conectores	• 2 conectores de 100 pines, 0.6mm pitch Hirose FX8C-100 para conectarse con la OEM Board • Conector Ethernet (RJ45) • Interfaz y conector MMC/SD • Conector JTAG • Pads para conector ETM
Interfaces	• Interfaz y conector USB OTG

	• Interfaz y conector USB device • Interfaz y conector USB host • Interfaz Full Modem RS232 en la UART #1 • Interfaz y conector CAN
Alimentación	• Suministro de energía vía USB o vía transformador externo (9-15VDC) • Condensador de 0.3F de backup para RTC
Expansión	• Conector para pantalla LCD en color QVGA (opciones de interfaces serie y paralelo)
Dimensiones	• 240x150 mm
Otros	• 5 pulsadores (4 vía I2C) • 5 LEDs (vía I2C) • 2 entradas analógicas • Conversor USB-Serie en la UART #0, y funcionalidad ISP • Pulsador y LED de Reset • Salida de Micrófono • Todas las señales de la tarjeta están disponibles en los conectores de expansión

Tabla 4. Características de la OEM Base Board

3.2 inch QVGA Touch Color LCD	
Características	**Descripción**
Tipo de pantalla	LCD
Interfaces	• Interfaz paralelo de 8 o 16 bits • Interfaz serie de 8 bits (SPI-like) (max. 10 MHz reloj)
Alimentación	2.5-3.3V
Dimensiones	3.2 pulgadas (diagonal)
Número de píxeles	240x320 RGB (tamaño QVGA)
Área visible	48.6 x 64.8 mm
Tamaño de un punto	0.2025 x 0.2025 mm
Número de colores	262K (para el modo de 18 bits), 65k (para el modo de 16 bits)
Temperatura de operación	-20 a + 70 grados centígrados
Otros	• Fondo iluminado por LED blanco, con control PWM • Bloque de pines 2x20 (100 mil) + bloque de 1x6 pines requerido para la interfaz del módulo.

Tabla 5. Características de la pantalla QVGA Touch Color LCD

5.1.2. Desarrollo

En este apartado se detalla el desarrollo llevado a cabo para la implementación de la aplicación que controla el módulo hardware [12].

La aplicación se ha desarrollado en su totalidad en lenguaje C, y funciona sobre un sistema operativo basado en una distribución Linux muy limitada y ligera, con el objetivo de consumir poca memoria.

El árbol de clases C que definen el diseño es el que se muestra en la figura 10.

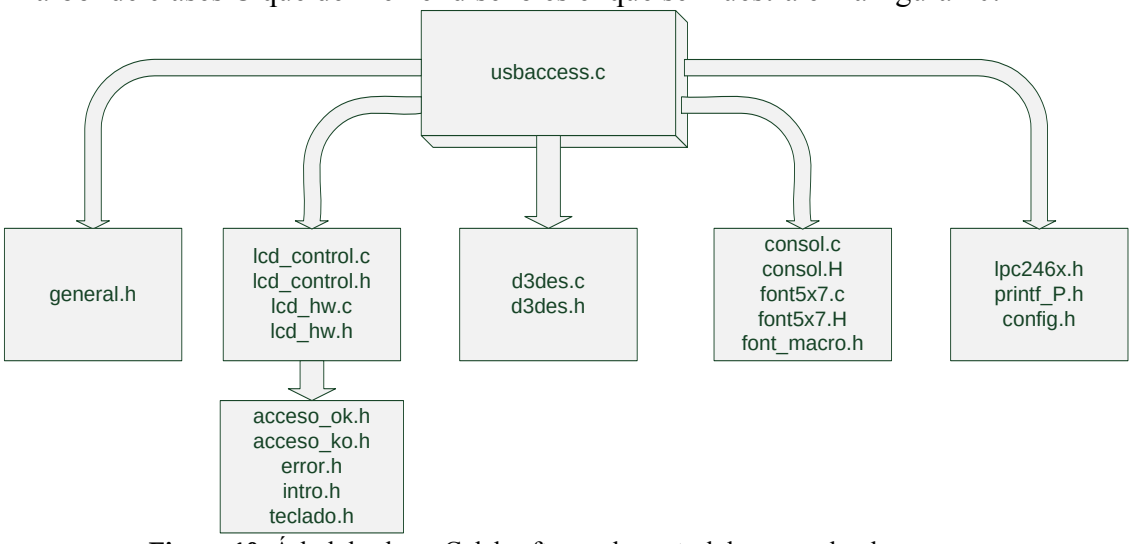


Figura 10. Árbol de clases C del software de control de acceso hardware.

El programa esta dividido en varias partes, cada una de las cuales se encarga de gestionar una actividad de la aplicación. Así, los módulos en los que está dividida son los siguientes:

1. Configuración general – Se encarga de inicializar las variables generales y configurar los pines del microprocesador en función de la aplicación.
2. Configuración específica – Se encarga de activar la configuración especifica de los distintos periféricos que maneja la aplicación.
3. Configuración de fuentes – Se encarga del manejo de las fuentes de texto, colores y formas que se mostrarán en el LCD.
4. Control del LCD – Se encarga de hacer toda la gestión de la pantalla LCD, desde la configuración de los pines, hasta la manipulación de imágenes.
5. Control de cifrado – Se encarga de gestionar las etapas de cifrado y descifrado del fichero llave del usuario.

A continuación se detalla la función de cada uno de los ficheros que componen la aplicación.

- **acceso_ko.h** - Contiene la descomposición en bytes de la imagen de confirmación negativa de la clave.
- **acceso_ok.h** - Contiene la descomposición en bytes de la imagen de confirmación positiva de la clave.
- **config.h** - Se encarga de inicializar los relojes, las pilas y definir los tamaños de los diferentes módulos de memoria (ROM, RAM, EEPROM).
- **consol.c** - Implementa las funciones relacionadas con el control de la consola, como las relacionadas con la impresión y captura de caracteres y de cadenas. Se utiliza para la represtación de datos en el programa terminal.
- **consol.h** - Contiene las definiciones de las funciones de consol.c.
- **des.c** - Implementa las funciones de cifrado y descifrado basadas en el algoritmo DES. Contiene funciones para cifrar y descifrar con DES y TripleDES, y para generar las claves.
- **des.h** - Contiene las definiciones de las funciones de d3des.c.
- **error.h** - Contiene la descomposición en bytes de la imagen de error que se muestra cuando no existe el fichero de clave en la memoria flash USB introducida.
- **font5x7.c** - Contiene una representación de todos los caracteres en formato de byte para poder representarlos en el display.
- **font5x7.h** - Contiene las declaraciones de las variables font5x7.c.
- **font_macro.h** - Contiene la definición de las macros para la representación de caracteres en el display.
- **general.h** - Este fichero define los tipos de datos que se manejan en la aplicación, a partir de tipos primitivos existentes en C.
- **intro.h** - Contiene la descomposición en bytes de la imagen de bienvenida al usuario con las instrucciones para iniciar la autenticación.
- **lcd_control.c** - Implementa las funciones que controlan el display, como la representación de figuras geométricas, de caracteres, de cadenas de texto o de imágenes. Además implementa la función que representa el teclado y gestiona las pulsaciones introducidas para recuperar la clave del usuario.
- **lcd_control.h** - Contiene las definiciones de las funciones de lcd_control.c.
- **lcd_hw.c** - Implementa todo el control de bajo nivel de la pantalla táctil, configurando los pines para activar cada una de las funcionalidades.
- **lcd_hw.h** - Contiene las definiciones de las funciones de lcd_hw.c.
- **lpc246x.h** - Este módulo implementa un controlador de dispositivo de caracteres Linux.
- **printf_P.h** - Fichero de cabecera para la familia de microprocesadores LPC246x. Este fichero es el conjunto de todas las definiciones hardware de los periféricos de esta familia de micros.

- **teclado.h** - Contiene la descomposición en bytes de la imagen del teclado que el usuario utilizará para introducir su clave.
- **usbaccess.c** - Es el módulo principal encargado de realizar las llamadas al resto de las funciones y gestionar cada uno de los estados de la aplicación.

5.1.3. Funcionamiento

A continuación se presentan una serie de diagramas UML, que explican el funcionamiento de la aplicación que se ejecuta sobre la tarjeta LPC2468 OEM Board.

Para iniciar el sistema de autenticación, el usuario tiene que introducir la llave USB en el puerto USB del equipo de la pared. En ese momento se procede a la autenticación del usuario. Para ello el sistema comprueba si la llave es correcta. En caso de serlo, se presenta una pantalla con un teclado donde el usuario debe introducir su clave haciendo uso de un lápiz. Si la clave es correcta el usuario tendrá acceso inmediato a la zona restringida.

La figura 11 representa la secuencia de estados por los que pasa la aplicación en función de la actividad del usuario.

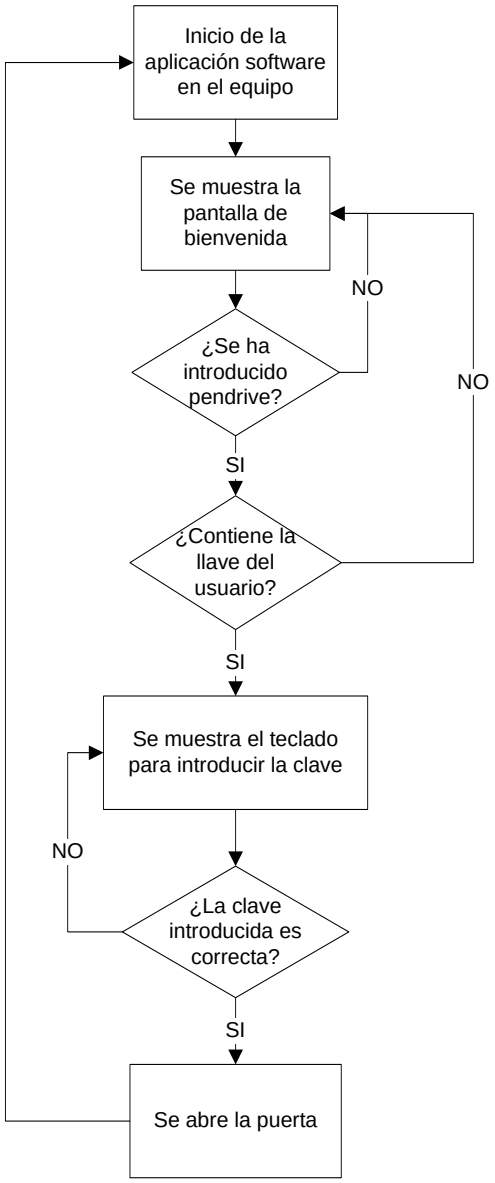


Figura 11. Diagrama de flujo del sistema de control de acceso a un área restringida

La figura 12 representa gráficamente la secuencia de ventanas que se muestran en función de cada uno de los estados de la aplicación.

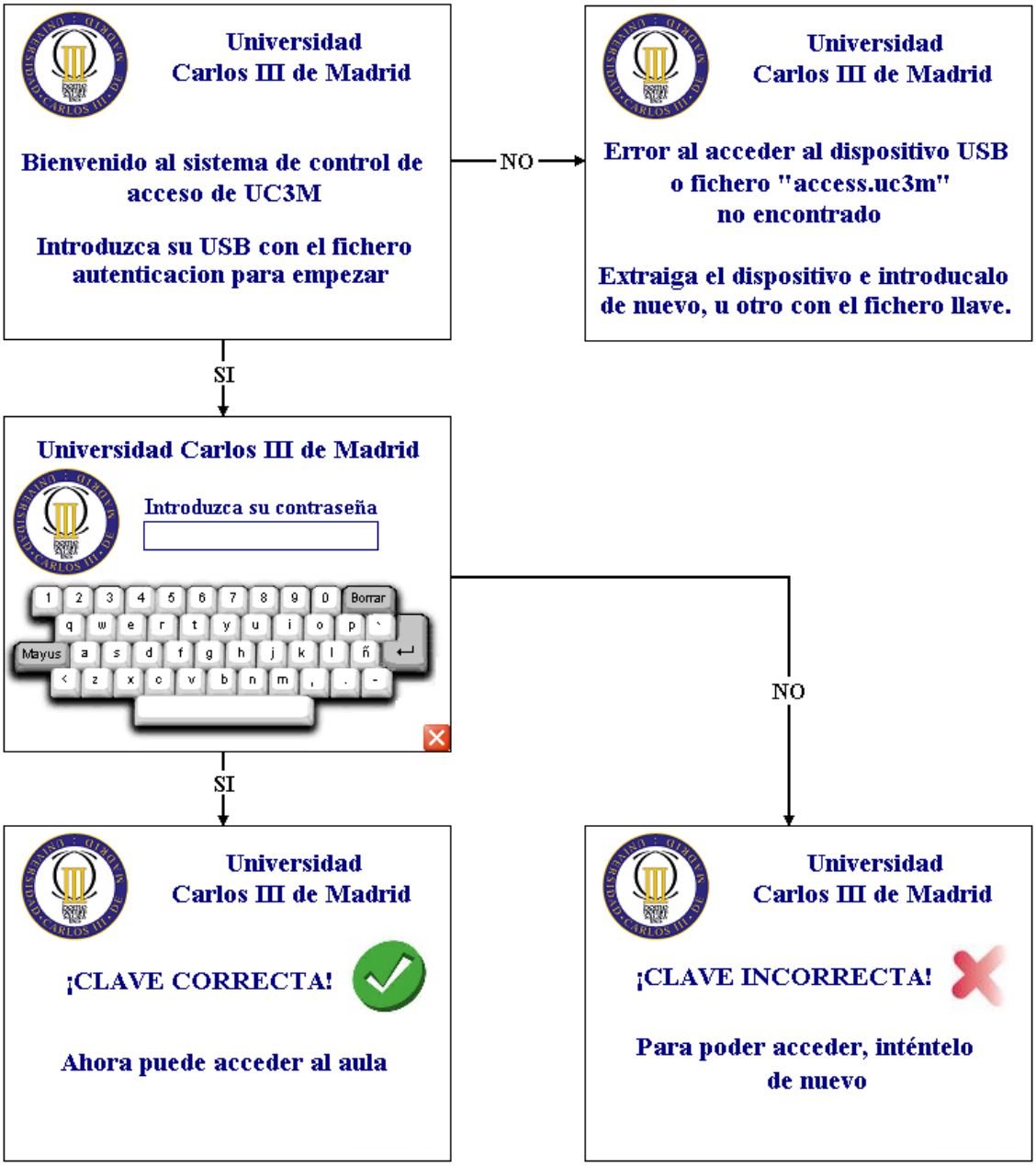


Figura 12. Diagrama de flujo gráfico del sistema de control de acceso a un área restringida

El diagrama de secuencia de la figura 13 representa el proceso mediante el cual, el usuario accede a un laboratorio o zona controlada.

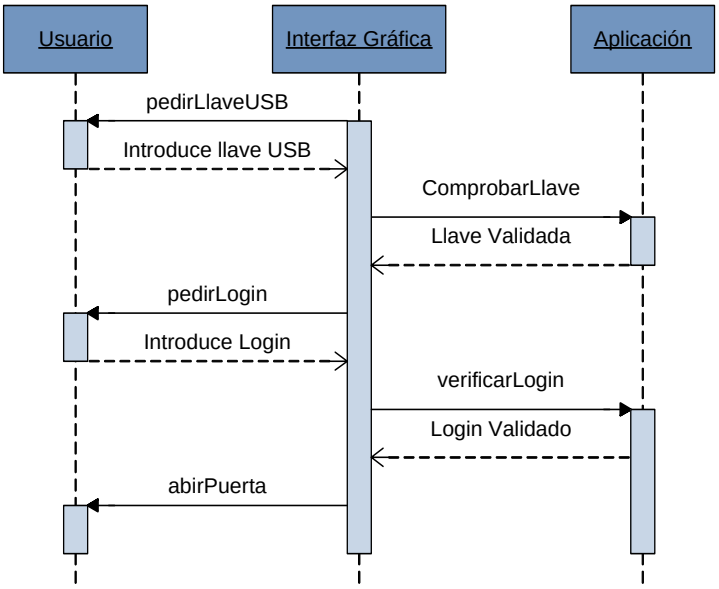


Figura 13. Diagrama de actividad para el acceso a una zona controlada.

El diagrama de secuencia de la figura 14 representa el proceso mediante el cual el usuario accede a un laboratorio o zona controlada con una contraseña incorrecta.

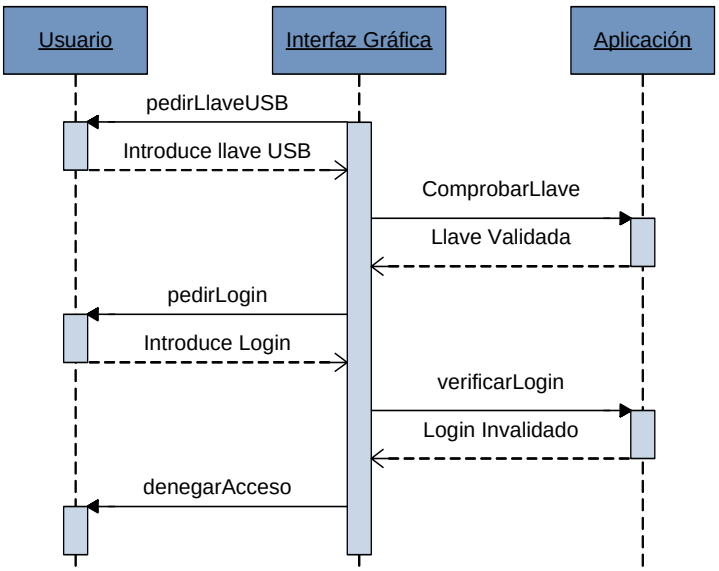


Figura 14. Diagrama de actividad para el acceso a una zona controlada con contraseña incorrecta.

5.2. Implementación de la aplicación en Java para el subsistema software

En este apartado se explicará como se ha realizado la implementación de la aplicación en Java [3] [5] para llevar a cabo el desarrollo de las distintas funciones del sistema de control de acceso por software, así como el API que se ha utilizado.

5.2.1. Diseño de la aplicación

Este subsistema es una aplicación software que controla el acceso a los equipos informáticos de los laboratorios, aulas u otras zonas restringidas. Se basa en una interfaz bloqueante que impide el uso de los equipos en el estado previo a la autenticación.

Esta aplicación ha sido desarrollada completamente en Java. La principal motivación de usar este lenguaje es por su carácter multiplataforma, permitiendo migrar dicha aplicación a otros sistemas operativos en el futuro, con gran sencillez. También se tuvieron en cuenta otras características como que es un lenguaje de programación muy extendido y completo, la versatilidad para el desarrollo de interfaces graficas, y por las bibliotecas de manejo de las interfaces USB.

Además del control de acceso, este software genera estadísticas en tiempo real en función de las aplicaciones que esté utilizando el usuario y almacena copias de dichos datos tanto en memoria flash USB del usuario, como remotamente en el equipo del administrador. De esta forma, basándose en los datos generados, se puede tener en cuenta el tiempo y el uso que cada usuario ha dado a los equipos.

El diagrama de bloques que describe la composición del programa esta representado en la figura 15.

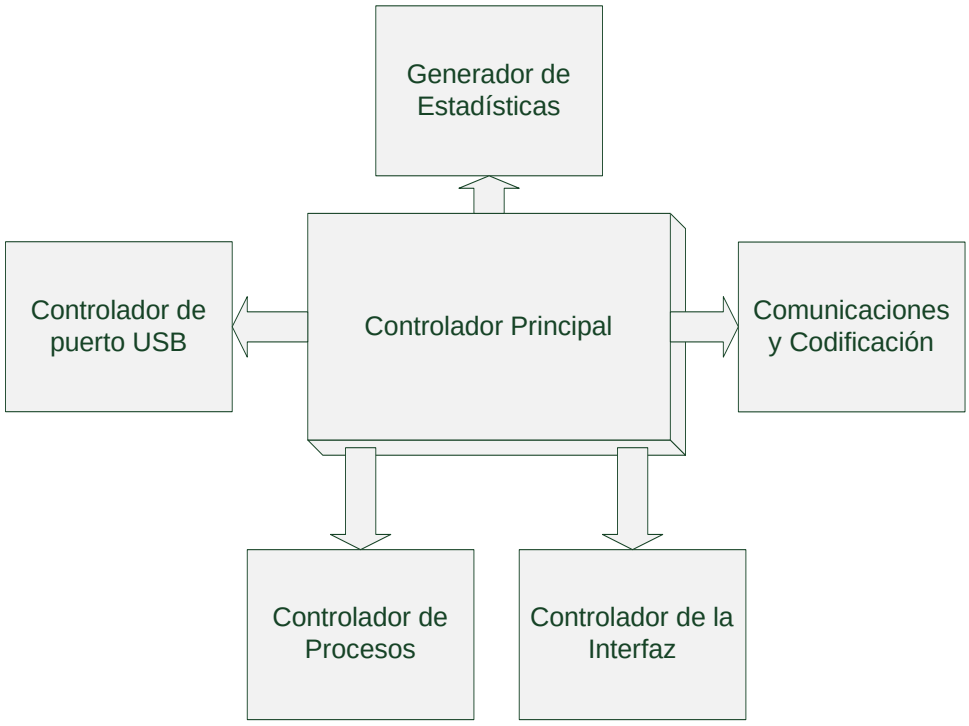


Figura 15. Diagrama de bloques del software de control de acceso software

Además, se ha diseñado otro módulo instalado en el equipo del administrador, y que se encarga de gestionar la conexión con los equipos de los usuarios y recibir los ficheros de estadísticas de éstos.

El diseño de este módulo es el que se muestra en la figura 16.

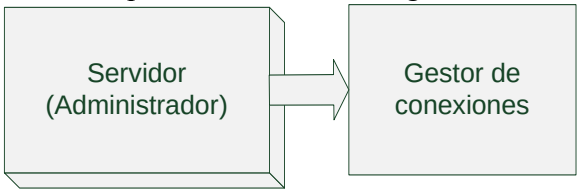


Figura 16. Diagrama de bloques módulo de comunicaciones del administrador.

El diseño físico del sistema se muestra en la figura 17, donde la línea roja representa la comunicación entre un equipo de usuario y el del administrador.

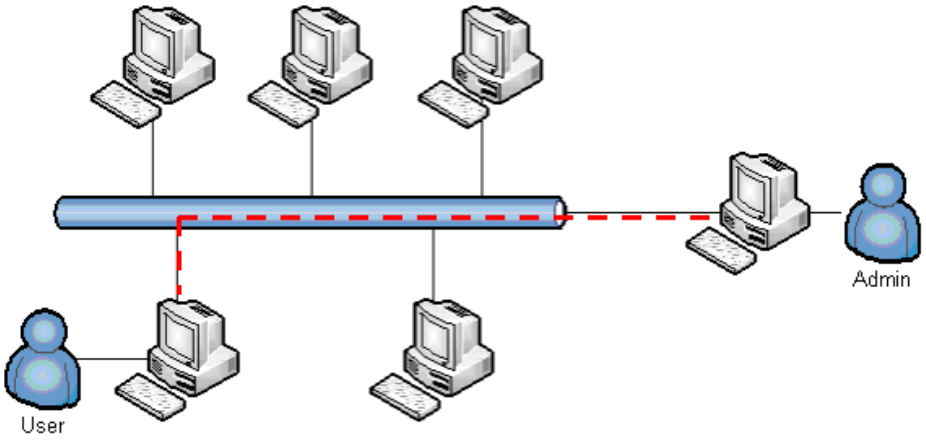


Figura 17. Esquema de comunicaciones entre el equipo del usuario y del administrador

Adicionalmente existe otra aplicación encargada de cifrar los ficheros de texto. También ha sido desarrollada en Java, aunque no forma parte directamente de la aplicación de control. Irá integrada en la página web de la universidad UC3M, de tal modo que uno de los enlaces de Plataforma Docente (Aula Global 2) dará acceso a ella, permitiendo a los usuarios generar su fichero llave cada vez que lo deseen. Esta aplicación interactuará con la base de datos para recolectar información acerca las asignaturas del usuario y generar el fichero en función de ellas.

El diseño de este módulo es el que se muestra en la figura 18.

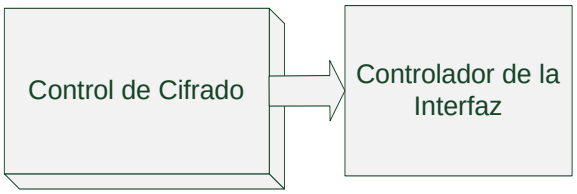


Figura 18. Diagrama de bloques de la aplicación de cifrado.

A continuación se presentan las distintas pantallas que se mostrarán para cada uno de los estados del software de control de los equipos de usuario.

En primer lugar se muestra una ventana que solicita al usuario introducir un dispositivo USB de almacenamiento con el fichero llave en su interior

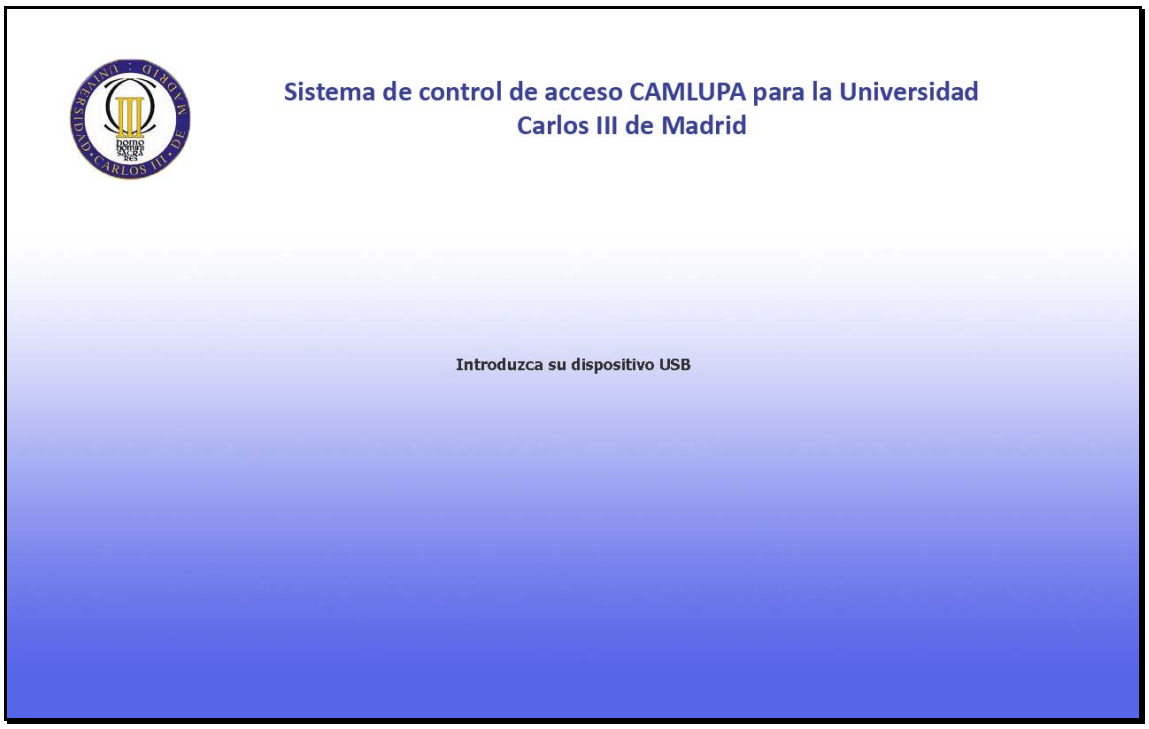


Figura 19. Pantalla de inicio de sesión en los equipos de los usuarios.

A continuación se presenta un formulario en el que el usuario deberá introducir su NIA y su contraseña. Realmente el NIA no es necesario puesto que se recupera automáticamente del fichero llave. Sin embargo es más cómodo para el usuario disponer de una interfaz parecida a la del resto de sistemas de acceso de la universidad (como por ejemplo Plataforma Docente (Aula Global 2)) con los que ya está familiarizado.

Figura 20. Pantalla de petición de login y password de usuario.

Además se dispone de una interfaz web que se encarga de generar los ficheros llave de cada uno de los usuarios, teniendo en cuenta las asignaturas en las que se ha matriculado. La interfaz de inicio es la que se presenta en la figura 21. Al final de la lista de opciones de la izquierda aparece un enlace llamado "Fichero Llave USB", sobre el que habrá que pulsar.

Figura 21. Pantalla de acceso al generador del fichero llave USB.

Una vez se haya accedido al enlace "Fichero Llave USB", se mostrará la siguiente pantalla en la que el fichero ha sido generado y podrá ser descargado directamente del enlace que se muestra en la figura 22..

Figura 22. Pantalla para la descarga de los ficheros llave USB.

La ventana de descarga típica de cualquier navegador facilitará la descarga de los ficheros a la ubicación deseada.

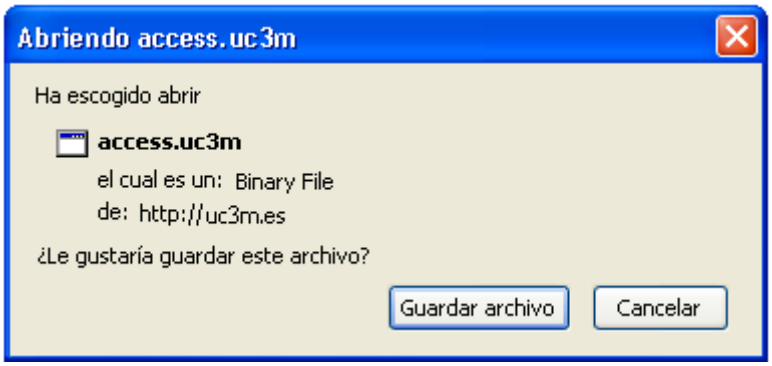


Figura 23. Pantalla de descarga de ficheros del navegador web.

5.2.2. API Java para la seguridad y criptografía

Para la implementación del programa ha sido necesario desarrollar una serie de clases, destinadas al cifrado y descifrado de ficheros con los datos de los usuarios para garantizar la seguridad. Para ello se han utilizado las bibliotecas de seguridad que proporciona Java, y que facilitan la integración de funciones criptográficas. [14]

El conjunto de clases de seguridad que proporciona Java puede dividirse en dos subconjuntos:

1. Clases relacionadas con el control de acceso y la gestión de permisos (`java.security`).
2. Clases relacionadas con la Criptografía (`javax.crypto`).

A partir del JDK (Java Developement Kit) 1.1 de Java se incluyen clases de acceso a funciones criptográficas de propósito general, conocidas como la Arquitectura Criptográfica de Java (JCA, Java Cryptography Architecture) y la Extensión Criptográfica de Java (JCE – Java Cryptography Extension). El JCA está formado por las clases básicas relacionadas con criptografía distribuidas con el JDK y se encuentra integrado en el paquete `java.security`. El paquete de extensión JCE proporciona el soporte para la encriptación y se encuentra integrado en el paquete `javax.crypto`.

Tanto la Arquitectura Criptográfica de Java como la Extensión Criptográfica de Java forman parten del API de Java, y por tanto no han sido desarrolladas para este proyecto. Por el contrario se han utilizado, para el desarrollo de la parte de seguridad de la aplicación.

5.2.2.1. Arquitectura Criptográfica de Java (JCA)

El JCA, como se ha comentado anteriormente, esta integrado dentro del paquete `java.security`, que está formado básicamente por un conjunto de clases abstractas e interfaces que encapsulan conceptos de seguridad como certificados, claves, resúmenes de mensajes y firmas digitales.

La primera versión del JCA apareció con la versión JDK 1.1, y en el se incluían las clases necesarias para facilitar el acceso y el desarrollo de la funcionalidad criptográfica para Java.

Existen distintas clases que pueden implementar los proveedores para proporcionar servicios criptográficos. Las tres primeras pertenecen a la primera versión del JCA, mientras que el JCA 1.2 amplía considerablemente el número de estas clases, y es al que pertenecen las restantes.

- **`KeyPairGenerator`** – Se emplea para crear claves públicas y privadas, así como pares de claves (pública/privada).
- **`MessageDigest`** – Proporciona la funcionalidad de algoritmos de resumen de mensajes como el MD5 y el SHA (función hash).
- **`Signature`** – Se utiliza para el firmado digital de mensajes y la verificación de firmas.

- **`AlgorithmParameterGenerator`** – Se utiliza para generar el conjunto de parámetros necesarios para usar un determinado algoritmo.
- **`AlgorithmParameters`** – Se emplea como representación opaca de los parámetros de un algoritmo.
- **`KeyFactory`** – Se utiliza para convertir claves (claves criptográficas opacas de tipo `Key`) en especificaciones de claves (representaciones transparentes de las claves) y viceversa.
- **`KeyStore`** – Esta clase representa una colección de certificados y claves en memoria.
- **`SecureRandom`** – Esta clase proporciona un generador de números pseudo-aleatorios criptográficamente seguro.

El paquete **`java.security`** esta formado por:

- **`java.security.spec`** – Fue introducido en la versión 1.2 del JCA, añade interfaces y clases que describen formatos de especificación de claves. Estas clases permiten crear claves de seguridad en Java basadas en parámetros generados por herramientas externas al JDK. Entre otras, existen clases para representar parámetros y claves de los algoritmos DSA y RSA y claves codificadas según la especificación del X.509.
- **`java.security.cert`** – Proporciona soporte para generar y usar certificados, así como leerlos de lugares como archivos JAR a través de las clases **`JarURLConnection`** y **`JarEntry`**. Además incluye clases e interfaces específicas para soportar certificados X.509. Las clases más importantes incluidas en paquete son:
  - **`CertificateFactory`** – Se emplea para generar certificados y listas de revocación (**CRL**).
  - **`Certificate`** – Clase abstracta para gestionar certificados. Es una clase para agrupar certificados de diferentes formatos pero usos comunes importantes, como por ejemplo funciones de codificación y verificación o datos como claves públicas.
  - **`CRL`** – Clase abstracta para gestionar distintos tipos de listas de revocación de certificados.
  - **`X509Certificate`** – Clase abstracta para representar certificados X.509. Proporciona un método estándar para acceder a los atributos de un certificado X.509.
  - **`X509Extension`** – Es una interfaz para las extensiones del formato X.509.
  - **`X509CRL`** – Clase abstracta para una lista de revocación de certificados X.509.
  - **`X509CRLEntry`** – Es una clase abstracta para las entradas de las listas de revocación.
- **`java.security.interfaces`** – La versión 1.1 del JCA incluía en este paquete interfaces para usar el algoritmo DSA. El JCA 1.2 amplió la clase para incluir interfaces para el algoritmo RSA. Las interfaces incluidas en paquete son:
  - **`DSAKey`** – Interfaz para una clave DSA pública o privada.
  - **`DSAKeyGenerator`** – Interfaz para un objeto capaz de generar pares de claves DSA.
  - **`DSAKey`** – Interfaz para una clave DSA pública o privada.

- **`DSAParams`** – Interfaz para un conjunto de parámetros específicos del claves de tipo DSA.
- **`DSAPrivateKey`** – Interfaz para una clave privada DSA.
- **`DSAPublicKey`** – Interfaz para una clave pública DSA.
- **`RSAPrivateCrtKey`**. Interfaz para una clave privada RSA, como se define en el estándar PKCS#1.
- **`RSAPrivateKey`** – Interfaz para una clave privada RSA.
- **`RSAPublicKey`** – Interfaz para una clave pública RSA.

- **`java.security.acl`** – Define soporte para listas de control de acceso que se pueden emplear para restringir el acceso a recursos de cualquier manera deseada. El paquete consta de interfaces y excepciones.

5.2.2.2. Extensión Criptográfica de Java (JCE)

La JCE es una extensión de la plataforma Java y proporciona implementaciones de algoritmos que permiten encriptar, generar claves, intercambiar claves y autentificar mensajes. La extensión complementa las interfaces e implementaciones de resumen y firmado de mensajes del JDK 1.2. El JCE está integrado dentro del paquete **`javax.crypto`**.

El paquete **`javax.crypto`** proporciona las clases e interfaces para realizar operaciones criptográficas. Incluye encriptación, generación de claves y acuerdo de claves y generación de códigos de autentificación de mensajes (MAC). El soporte para encriptación incluye algoritmos de cifrado simétricos, asimétricos, por bloques y de flujo. La mayoría de clases incluidas en este paquete se basan en proveedores, las clases sólo definen el API.

Al igual que el JCA, el modelo empleado por el JCE está basado en el uso de *proveedores*. El paquete está formado por:

- **`javax.crypto`** → Es el paquete principal, y de él derivan los otros dos subpaquetes. Está formado por clases que representan los conceptos de cifrado, acuerdos de claves y códigos de autentificación de mensajes y sus clases de interfaz de proveedor (SPI). Las clases más importantes incluidas en el paquete **`javax.crypto`** son:
  - **`Cipher`** – Representa un algoritmo de cifrado.
  - **`KeyAgreement`** – Proporciona la funcionalidad de un protocolo de acuerdo o intercambio de claves.
  - **`KeyGenerator`** – Proporciona las funciones de un generador de claves simétricas.
  - **`Mac`** – Proporciona las funciones de un generador de códigos de autentificación de mensajes.
  - **`SecretKeyFactory`** – Representa un generador de claves secretas.
- **`javax.crypto.spec`** – Es uno de los subpaquetes derivados de **`javax.crypto`**. Está formado por varias clases de especificación de claves y de parámetros de algoritmos.
- **`javax.crypto.interfaces`** – Es el otro subpaquete que conforma el **`javax.crypto`**. Presenta las interfaces de las claves empleadas en los diferentes tipos de algoritmos.

5.2.3. API de Java para control de puertos USB

El control de los puertos USB ha sido uno de los puntos más importantes para el desarrolló de la aplicación de control de acceso a los equipos, puesto que dicho acceso se realizaba a través de una memoria flash con interfaz USB.

Las bibliotecas para el control de puertos USB desarrolladas para Java han facilitado la utilización de este tipo de interfaces.

El proyecto jUSB fue creado por Mojo Jojo y David Brownell en Enero del 2000 [17]. Se desarrolló con el objetivo de proveer un conjunto gratuito de APIs Java para acceder a dispositivos USB en plataformas Windows y Linux. Este conjunto de clases es distribuido bajo la licencia Lesser GPL (LGPL), lo que permite su utilización en proyectos propietarios, o gratuitos.

Provee acceso multi-hilo a múltiples dispositivos USB físicos, y soporta tanto dispositivos nativos como remotos. Los dispositivos con múltiples interfaces pueden accederse desde múltiples aplicaciones (o *device drivers*) simultáneamente, permitiendo que cada aplicación (o driver) solicite una interfaz diferente. Otra característica es que soporta distintos modos de tranferencia de datos (*control transfers, bulk transfers,* e *interrupt transfers*).

El API jUSB incluye los siguientes paquetes:

- **usb.core** – Este paquete es el núcleo. Permite a las aplicaciones Java acceder a dispositivos USB desde hosts USB.
- **usb.linux –** Este paquete contiene la implementación Linux de un objeto **usb.core.Host**, soporte **bootstrapping**, y otras clases ofreciendo soporte USB para Linux. Esta implementación accede a dispositivos USB a través del dispositivo virtual USB definido a nivel de file system (usbdevfs).
- **usb.windows –** Este paquete contiene la implementación Windows de un objeto **usb.core.Host**, soporte **bootstrapping**, y otras clases ofreciendo soporte USB para Windows. Esta implementación se encuentra aún en etapas iniciales de desarrollo.
- **usb.remote –** Este paquete es una versión remota de la API **usb.core**. Incluye un proxy RMI y una aplicación daemon, que permite a las aplicaciones Java acceder a dispositivos USB ubicados en un ordenador remoto.
- **usb.util –** Este paquete provee utilidades para descargar firmware de dispositivos USB, volcar el contenido del sistema USB en XML, y convertir un dispositivo USB con datos exclusivamente I/O en un socket.
- **usb.devices –** Este paquete opcional se usa para acceder a una determinada variedad de dispositivos USB. Estas clases fueron programadas especialmente para simplificar el proceso de acceder determinados dispositivos USB. Fueron construidas utilizando las clases del paquete **usb.core**, y funcionarán sobre cualquier sistema operativo soportado por jUSB.
- **usb.view –** Este paquete opcional provee un simple navegador jerárquico USB basado en Swing.

5.2.4. Desarrollo

En este apartado se detalla el desarrollo llevado a cabo para la implementación de la aplicación que controla el módulo software.

La aplicación se ha desarrollado en su totalidad en lenguaje Java y utilizando el entorno de desarrollo Eclipse (*véase sección 3.1.1, Herramienta de trabajo Eclipse*). Adicionalmente se ha utilizado una pequeña utilidad desarrollada en C, para la gestión de los procesos del sistema operativo Windows[2].

A continuación se muestran los diagramas de bloques de cada una de las aplicaciones que integran el sistema de control de acceso software.

5.2.4.1. Aplicación Principal

La aplicación principal irá instalada en los equipos susceptibles de ser utilizados por los usuarios, y se encargará de controlar el acceso y generar las estadísticas de uso.

El árbol de clases Java que definen el diseño es el que se muestra en la figura 24.

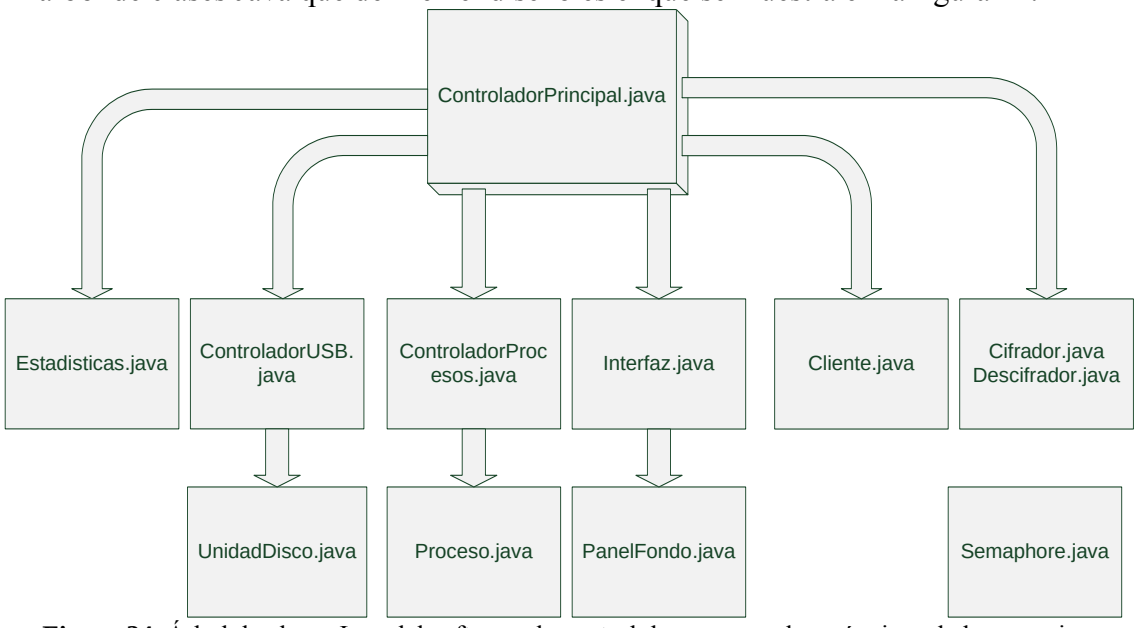


Figura 24. Árbol de clases Java del software de control de acceso en las máquinas de los usuarios.

El programa esta dividido en varias partes, cada una de las cuales se encarga de gestionar una actividad de la aplicación. Así, los módulos en los que está dividida son los siguientes:

1. Controlador principal – Se encarga de realizar toda la gestión de la aplicación y de hacer las llamadas a cada uno de los módulos para activar cada uno de los subprocesos del sistema.
2. Control de estadísticas – Se encarga de crear el hilo que administrar la información recabada sobre la actividad que realiza el usuario.

[2] Esta aplicación se encarga de monitorizar todos los procesos que se ejecutan en el sistema.

3. Control USB – Se encarga del manejo y control de los eventos en las interfaces USB del equipo, así como de la comunicación entre dicho equipo y los dispositivos que se conecten mediante USB.
4. Control de Procesos – Se encarga de crear el hilo que controla la actividad del usuario monitorizando cada uno de los procesos que tenga activos en el equipo.
5. Control de interfaz – Se encarga de gestionar la interfaz que se muestra al usuario en cada uno de los posibles estados. Define los elementos que deben mostrarse y las posiciones que deben ocupar en la pantalla.
6. Control del cliente – Se encarga de llevar a cabo la comunicación con el equipo del administrado mediante el uso de sockets. Es la parte que la aplicación utiliza para mandar los datos de cada usuario al administrador.
7. Control de cifrado – Se encarga de llevar a cabo el procesos de cifrado de los datos enviados al servidor y del descifrado de los ficheros de claves de cada usuario, para posteriormente poder autenticarle.

A continuación se detalla el diseño por clases llevado a cabo para el desarrollo de la aplicación principal.

- **Cifrador.java –** Esta clase contiene los métodos correspondientes al sistema de cifrado. Se encarga de cifrar el fichero con los datos personales y académicos del usuario. Se utiliza cifrado simétrico (*véase Anexo 1*), y en concreto el algoritmo DES (*Data Encryption Standard*) en el modo ECB (*Electronic Code Book*) con *PKCS5Padding* para el relleno de los bloques finales.
- **Cliente.java –** Esta clase es la encargada de crear un objeto cliente que establece una comunicación basada en sockets con el objeto servidor creado en la máquina del administrador. A través de este canal de comunicación, el administrador recibe todos los datos de uso de las máquinas de todos los usuarios del sistema.
- **ControladorPrincipal.java –** Se trata del hilo principal. Esta clase es la encargada de manejar y realizar las llamadas a todos los demás objetos del programa.
- **ControladorUSB.java –** Esta clase es la encargada de gestionar el puerto USB. Implementa la interfaz de *USBListener*, para poder manejar los eventos que se produzcan en estos puertos. Crea un nuevo hilo para realizar la gestión concurrentemente con el resto de la aplicación.
- **ControlProcesos.java –** Esta clase se encarga de controlar los procesos que se están ejecutando en el sistema operativo. Crea un nuevo hilo para realizar la gestión concurrentemente con el resto de la aplicación. Gracias a esta gestión se puede llevar un control de los programas que esta ejecutando el usuario, y por tanto filtrar aquellos relacionados con sus asignaturas. Esta clase utiliza una pequeña aplicación, llamada *tasklist*, que se encarga de recopilar información sobre los procesos en ejecución.
- **Descrifrador.java –** Esta clase contiene los métodos correspondientes al sistema de descifrado. Se encarga de, una vez recuperado el fichero de usuario (llave), descifrarla para contrastar las contraseñas. Se utiliza cifrado simétrico (*véase Anexo 1*), y en concreto el algoritmo DES (*Data Encryption Standard*) en el modo ECB (*Electronic Code Book*) con *PKCS5Padding* para el relleno de los bloques finales.

- **Estadisticas.java –** Esta clase se encarga de contabilizar el tiempo que se esta ejecutando cada uno de las aplicaciones relacionadas con las asignaturas del usuario. Además genera un fichero donde va almacenando todo la información recopilada, que luego se mandará al USB del usuario y al equipo del administrador. Este fichero es el que posteriormente se tendrá en cuenta para realizar la evaluación del trabajo del usuario. También genera un nuevo hilo para realizar la gestión concurrentemente con el resto de la aplicación.
- **Interfaz.java –** Esta clase es la encargada de representar las distintas interfaces graficas de la aplicación. Crea una interfaz en función del estado de la aplicación. Su funcionamiento está basado en la utilización de bibliotecas de *Java Swing* (biblioteca para la creación de interfaces gráficas). Todas las interfaces que genera son bloqueantes, por lo que mientras que estén activas, el usuario no podrá realizar ninguna operación que no sea sobre ellas. Esto permite gestionar el control de acceso, de tal forma que hasta que el usuario no se autentique, la aplicación no libera el equipo.
- **PanelFondo.java –** Esta clase funciona como apoyo a la clase **Interfaz.java**. Se encarga de crear objetos con una imagen de fondo, de forma que luego puede representarse sobre las interfaces graficas.
- **Proceso.java –** Es una clase que genera objetos de tipo proceso, donde se almacena toda la información referente a un proceso determinado. Sirve como apoyo a la clase **CotrolProcesos.java**.
- **Semaphore.java –** Esta clase es la representación de un objeto de tipo semáforo, utilizado para la gestión de la concurrencia. Todo el control concurrente esta basado en el uso de semáforos, sea cual sea el paradigma que haya que resolver. Se tomó la decisión de implementar este mecanismo puesto que además de ser versátil, es muy ligero en cuanto a programación y gestión se refiere.
- **UnidadDisco.java –** Esta clase crea objetos de tipo unidad de disco, donde se almacena toda la información referente a cada una de las unidades de disco detectadas. Sirve como apoyo a la clase **ControlUSB.java**.

5.2.4.2. Aplicación Servidor

Esta aplicación irá instalada en las máquinas de los administradores. Su función es la de recibir los ficheros de estadísticas de los usuarios de forma periódica y de almacenarlos convenientemente con el fin de que luego puedan ser recuperados y procesados de forma sencilla.

El árbol de clases Java de la aplicación que gestiona el servidor instalado en el equipo del cliente es el que se muestra en la figura 25.

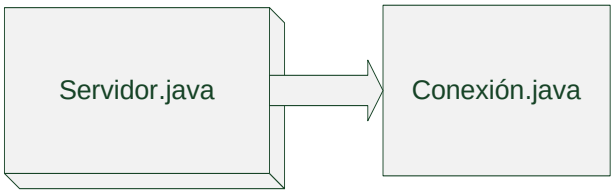


Figura 25. Diagrama de bloques de la aplicación servidor.

El programa esta dividido en varias partes, cada una de las cuales se encarga de gestionar una actividad de la aplicación. Así, los módulos en los que está dividida son los siguientes:

1. Controlador principal – Se encarga de iniciar las comunicaciones con cada uno de los equipos y de gestionar y administrar los datos recibidos.
2. Control de conexiones – Se encarga de crear los sockets para la comunicación con un equipo.

A continuación se detalla el diseño por clases llevado a cabo para el desarrollo de la aplicación servidor.

- **Servidor.java** – Es el encargado de iniciar las comunicaciones con el resto de equipos de los usuarios, además de gestionar y administrar los datos recibidos de ellos, facilitando las posteriores consultas de dichos datos.
- **Conexión.java** – Se encarga de crear e interconectar un socket de comunicaciones con un equipo de tipo cliente.

5.2.4.3. Aplicación Cifrado

Esta aplicación forma parte de la plataforma web de la universidad. Permitirá al usuario generar cuantas veces desee sus ficheros de acceso, incluso permitiéndole tener distintas claves. Esta última característica no altera el funcionamiento del sistema pero refuerza enormemente la seguridad.

El árbol de clases Java de la aplicación utilizada para cifra los ficheros de usuario son los representados en la figura 26.

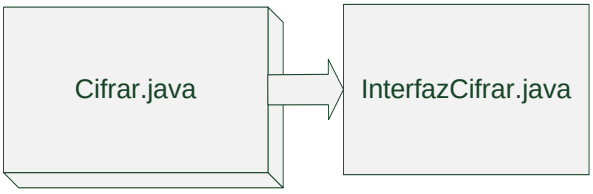


Figura 26. Diagrama de bloques (izquierda) y árbol de clases (derecha) de la aplicación de cifrado.

El programa esta dividido en varias partes, cada una de las cuales se encarga de gestionar una actividad de la aplicación. Así, los módulos en los que está dividida son los siguientes:

1. Control de cifrado – Se encarga de implementar el algoritmo de cifrado, en este caso DES, para poder generar el fichero de acceso del usuario en función de sus datos.
2. Interfaz de cifrado – Implementa la interfaz grafica para que el usuario pueda, de una manera cómoda y sencilla, generar automáticamente los ficheros de acceso al sistema.

A continuación se detalla el diseño por clases llevado a cabo para el desarrollo de la aplicación principal.

Las clases que forman el software para cifrar los ficheros llave de los usuarios son:

- **Cifrar.java –** Esta clase crea un objeto de tipo cifrador que contiene los métodos necesarios para llevar a cabo el cifrado de ficheros de texto.
- **InterfazCifrar.java –** Esta clase crea la interfaz grafica para poder seleccionar el fichero de texto a cifrar. Sirve como apoyo a la clase **Cifrar.java**.

5.2.5. Funcionamiento

A continuación se presentan una serie de diagramas UML, que explican el funcionamiento de la aplicación que corre en los equipos a los que tienen acceso los usuarios.

Para iniciar el sistema de autenticación, el usuario tiene que introducir la llave USB en el puerto USB del equipo (PC). En ese momento se procede a la autenticación del usuario. Para ello el sistema comprueba si la llave es correcta. En caso de serlo, se presenta un formulario donde debe introducir su clave haciendo uso del teclado del equipo. Si la clave es correcta el usuario tendrá acceso inmediato su sesión personal dentro del equipo.

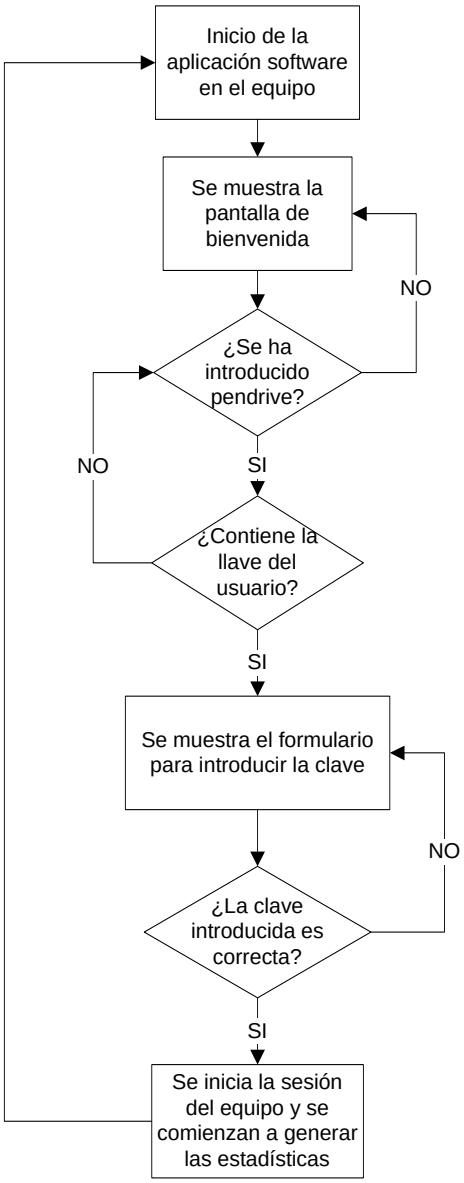


Figura 27. Diagrama de flujo del sistema de control de acceso a los equipos informáticos

El diagrama de secuencia de la figura 28 representa el proceso mediante el cual el usuario accede a su sesión en uno de los equipos preparados para dicho propósito.

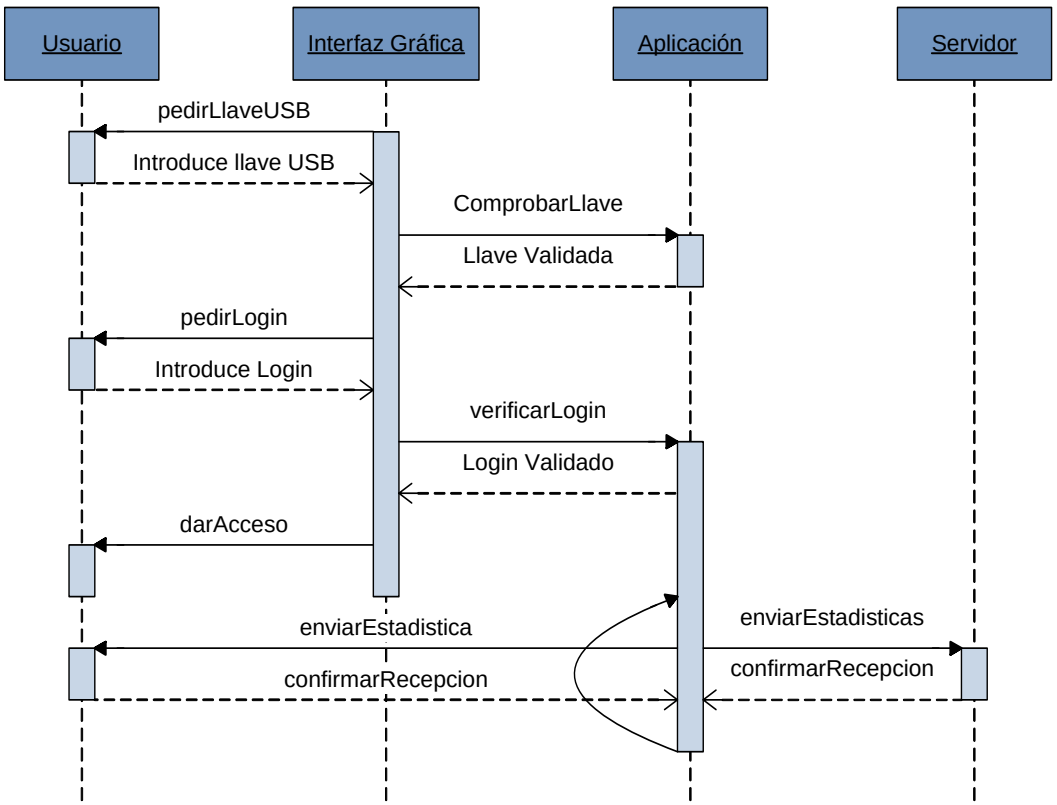


Figura 28. Diagrama de actividad para el acceso a un equipo del laboratorio.

El diagrama de secuencia de la figura 29 representa el proceso mediante el cual el usuario intenta acceder a su sesión en uno de los equipos del laboratorio con una contraseña incorrecta.

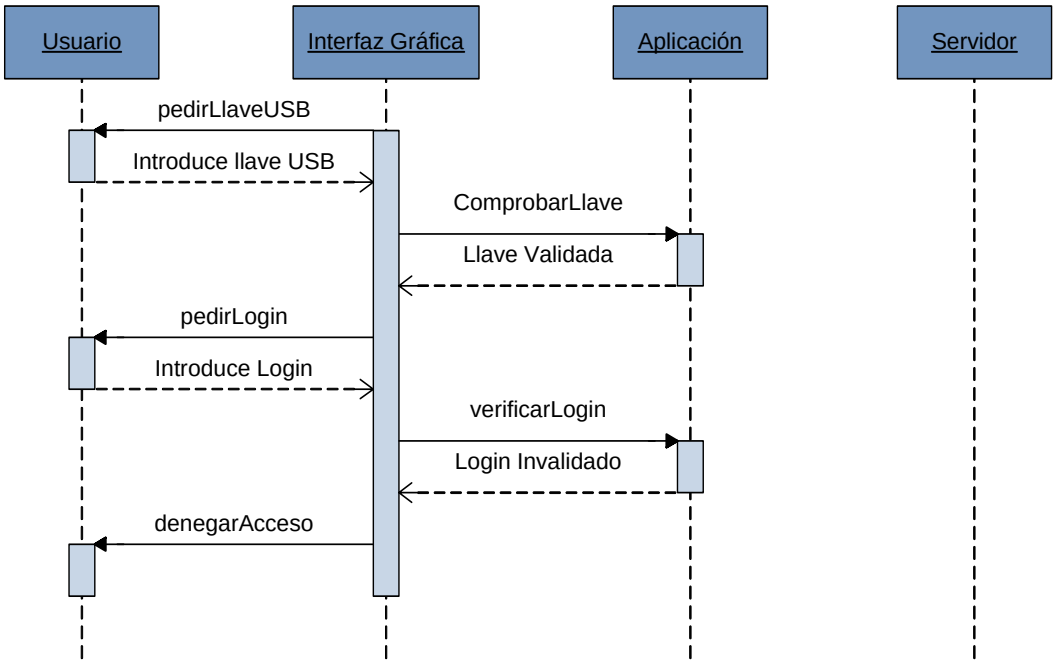


Figura 29. Diagrama de actividad para el acceso con clave incorrecta a un equipo del laboratorio.

La figura 30 representa el diagrama de secuencia del módulo de cifrado. Se considera que está integrado dentro de la aplicación web Plataforma Docente (Aula Global 2) de la universidad Carlos III de Madrid.

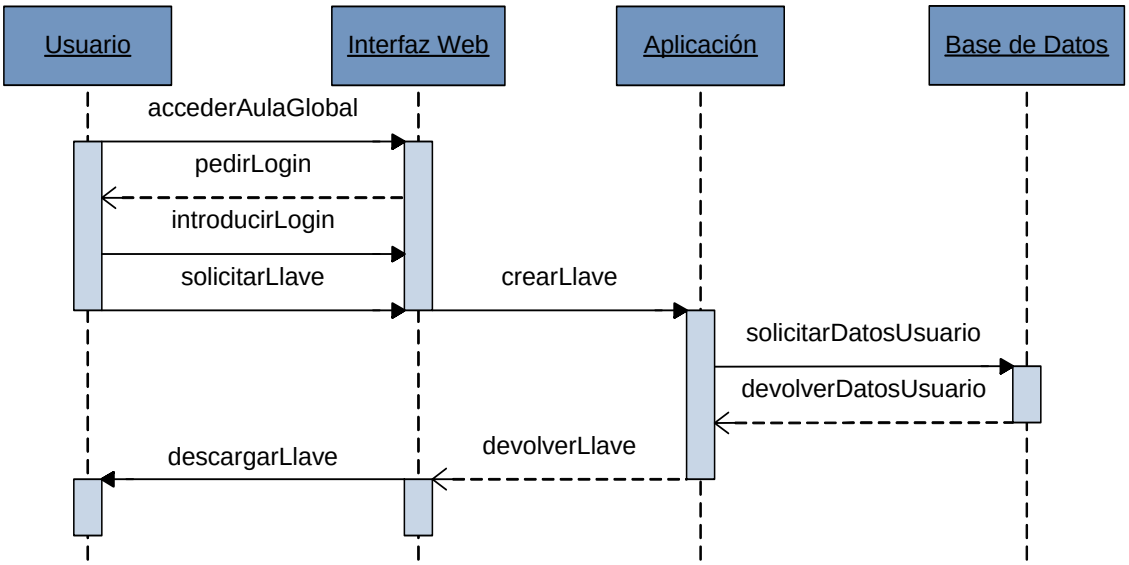


Figura 30. Diagrama de actividad para la solicitud del fichero llave.

6. Planificación de la instalación

En esta sección se describe el proceso que se ha de seguir para la implantación del sistema en un entorno real, y la configuración necesaria de cada uno de los equipos.

6.1. Configuración del equipo hardware (LPC2468 OEM Board)

La tarjeta LPC2468 OEM Board tiene distintos modos de configuración según la funcionalidad que se desee implementar o la actividad que se quiera realizar.

A continuación se muestran los diagramas de flujo para cada tipo de instalación.

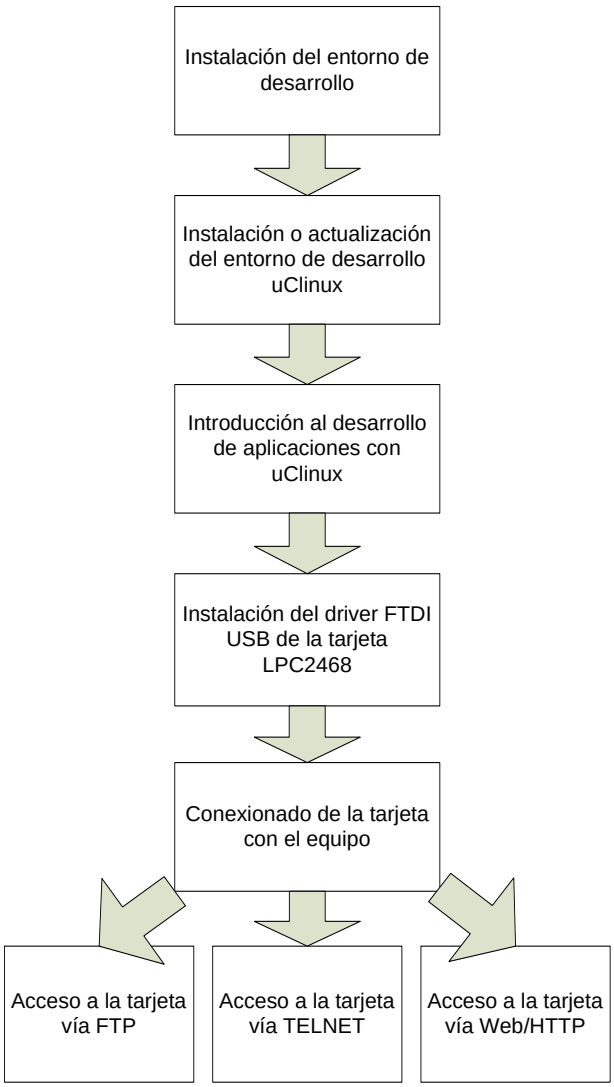


Figura 31. Secuencia de pasos para el desarrollo de aplicaciones

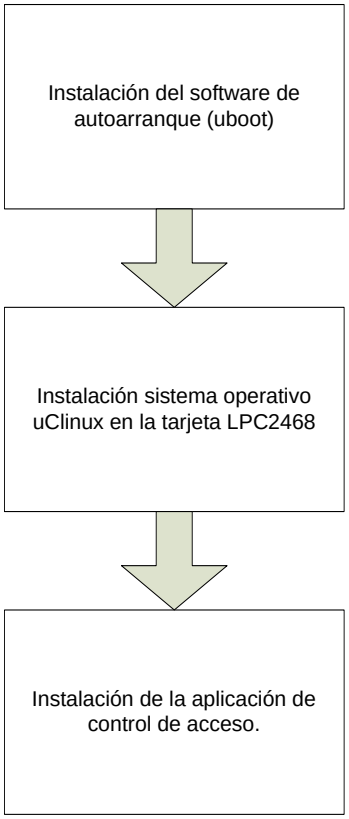


Figura 32. Secuencia de pasos para la instalación de la aplicación de control hardware

En los siguientes apartados se desarrolla en detalle cada uno de los pasos de los diagramas mostrados anteriormente.

6.1.1. Instalación del entorno de desarrollo

Esta sección es una introducción al SDK de la tarjeta LPC2468 OEM Board, incluyendo los detalles de la instalación.

El SDK de la tarjeta LPC2468 OEM Board se distribuye como una máquina virtual que tiene que ser ejecutada con el software VMWare Player, que es un producto libre de VMWare. Este software es compatible para la mayor parte de las versiones de Windows y de Linux. Para el resto de sistemas para los que no es compatible, se puede descargar el código fuente y compilarlo.

Las razones para utilizar VMWare Player son las siguientes:

- El SDK necesita un entorno de trabajo basado en Linux, algo que no todos los usuarios tienen instalado.
- La máquina virtual no necesita ninguna modificación ni configuración, sólo la de instalar el software VMWare Player.
- El SDK será y se comportará igual independientemente del sistema operativo sobre el que se ejecute.

El software VMWare Player se encuentra en directorio SDK/VMWarePlayer del CD, para Windows y para Linux. Es la versión 2.5.2, pero se puede descargar una versión actualizada de página web del desarrollador http://www.vmware.com/download/player.

Para realizar la instalación hay que seguir los siguientes pasos:

1. Tras haber descargado el programa VMWare Player, hay que hacer doble clic sobre el icono del programa y seguir las instrucciones de la instalación.
2. Una vez instalado VMWare Player, hay que descomprimir la máquina virtual con el SDK que correrá sobre él. Dicha máquina se encuentra en el directorio **SDK/VirtualMachines** del CD. Hay dos versiones de esta máquina:
   2.1. EADevEnv_v2_1.exe – Esta máquina es la que contiene la versión original de todos los programas, tal y como la distribuye EmbeddedArtists. Contiene la distribución del entorno de desarrollo de 2005 (uClinux-dist-20051014).
   2.2. EADevEnv_v2_1_uc3m.exe – Es una versión modificada de la anterior, con la distribución del entorno de desarrollo del 2007 (uClinux-dist-20070130) (*véase sección 6.1.2, Instalación o actualización del entorno de desarrollo uClinux*).
   Se selecciona la que corresponda y se descomprime en la ubicación deseada.
3. A continuación hay que ejecutar el fichero llamado **EA-dev.vmx** que se encuentra dentro de la ubicación donde se haya descomprimido el entorno y aparecerá una ventana como se indica en la figura 33.

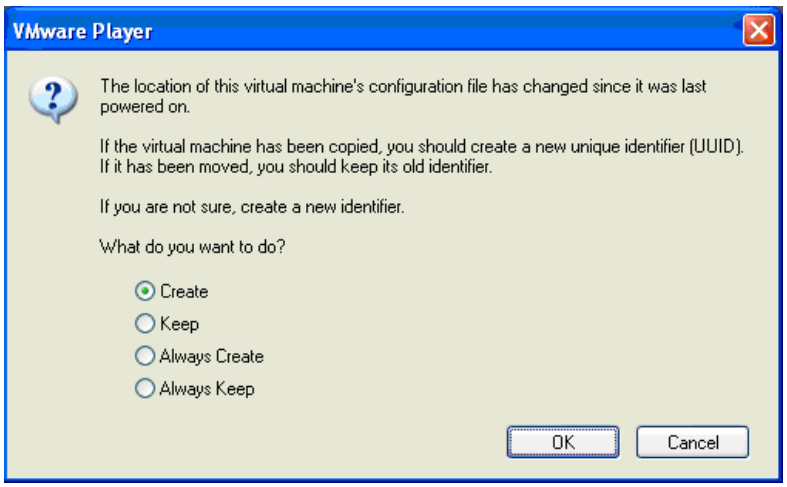


Figura 33. Ventana de configuración de la máquina virtual

4. Se selecciona 'Create' y el botón OK para generar un nuevo identificador para esta máquina virtual (esta ventana sólo aparece la primera vez que se inicie el SDK). La máquina virtual se autenticara automáticamente como el usuario por defecto. Cuando este proceso termine se mostrará el escritorio de la distribución de Linux como se ve en la imagen 34.

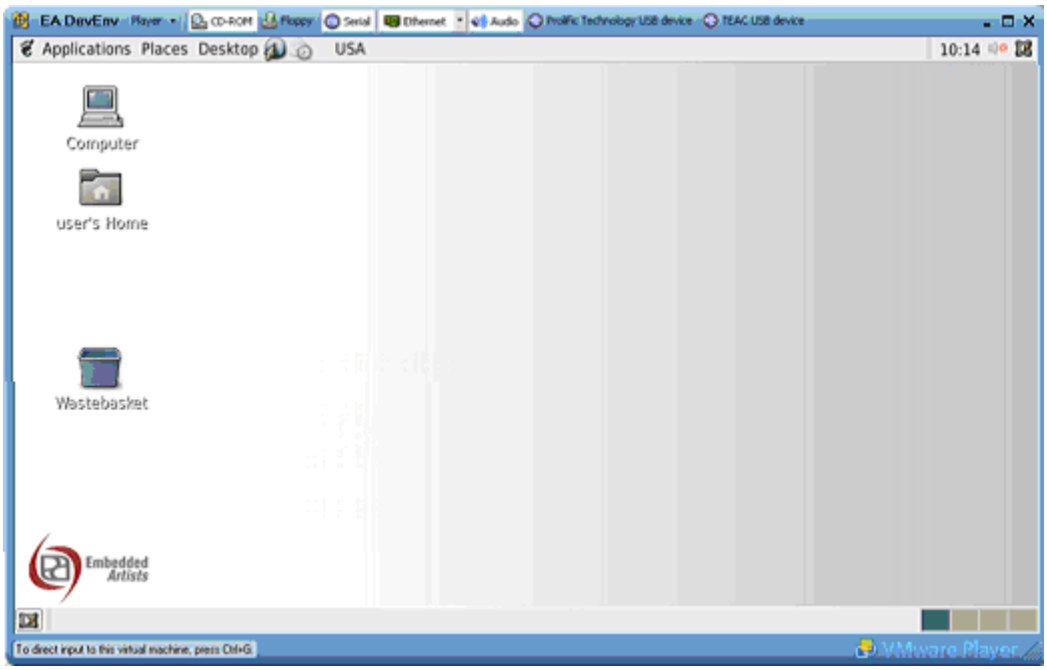


Figura 34. Escritorio de la máquina virtual de Linux

5. Para apagar la máquina virtual hay que seleccionar Desktop→Shutdown

NOTA: La contraseña para acceder al modo root es 'root'

6.1.2. Instalación o actualización del entorno de desarrollo uClinux

En esta sección se describe como actualizar el SDK para aplicar los nuevos parches, o como instalarlo desde el comienzo si ya se dispone de una distribución de Linux y/o no se desea utilizar VMWare Player.

Además describe cúales son los pasos necesarios para instalar el compilador cruzado del ARM y las herramientas relacionadas, y como construir el kernel y el sistema de ficheros inicial de uClinux.

Este entorno de desarrollo sólo se puede instalar bajo una distribución de Linux, y nunca bajo Cygwin/Windows.

Los requisitos necesarios para la instalación son:

- Linux (libc 2.3 o superior)
- GCC (3.4 o superior)
- Emulador de terminal (por ejemplo minicom)
- Cliente FTP
- Servidor NFS
- tar, bunzip2, gunzip, patch

La instalación del SDK requiere seguir los siguientes pasos:

1. Instalar paquete **arm-elf toolchain** (como root). Este programa se puede encontrar en la ubicación **SDK/Instalacion/arm-elf-tools-20061213.sh** del CD de software del proyecto. Las herramientas se instalarán en /user/local/bin y tendrán el prefijo arm-elf.

```
su
  Password: root
cd /
sh <downloadpath>/arm-elf-tools-20040427.sh
exit
```

2. El siguiente paso es instalar la distribución oficial de uClinux. Esta distribución es una colección de aplicaciones y bibliotecas portadas a uClinux. También contiene configuraciones especificas de distribuidores (vendors). Sin embargo la configuración y los drivers de Embedded Artists AB no son parte de la distribución oficial y por lo tanto han de ser instalados a parte.

   Esta distribución se encuentra en **SDK/Instalacion/uClinux-dist-20070130.tar.gz**. si se desea descargar una versión más actualizada habría que visitar la página www.uclinux.org.

```
cd <installation path>
tar -xzvf download_path/uClinux-dist-20070130.tar.gz
```

   En este paso se creará un nuevo directorio llamado uClinux-dist que contendrá todos los ficheros de la distribución.

3. Se cambia el directorio a uClinux-dist y se borran los subdirectorios con el nombre **linux-2.x.x**, ya que en esa ubicación irá instalado el nuevo núcleo.

```
cd uClinux-dist
rm -r linux-2.*
```

4. Ahora hay que instalar el kernel. La versión 2.6.21 del kernel se encuentra ubicada en **SDK/Instalacion/linux-2.6.21.tar.gz**. Nuevas versiones se pueden descargar directamente de www.kernel.org.

```
tar -jxvf <downloadpath>/linux-2.6.21.tar.bz2
mv linux-2.6.21 linux-2.6.x
gunzip -c your_path/ea-uClinux-081020.diff.gz | patch -p1
```

5. A continuación hay que aplicar el parche para la distribución de Embedded Artists AB

```
gunzip -c yourpath/uC-20051014-linux-2.6.11.8-ea1.diff.gz | patch –p1
```

6. Se selecciona la configuración de la tarjeta accediendo al menú gráfico de configuración. Una vez configurado, se compila para aplicar los nuevos cambios.

```
make menuconfig
```

```
Select Vendor/Product Selection
Select EmbeddedArtists
Select LPC2478OEM_Board or LPC2468OEM_Board
Exit
Exit
Yes
```

```
make dep
make
```

6.1.3. Introducción al desarrollo de aplicaciones con uClinux

El lenguaje de programación usado para el desarrollo de aplicaciones para la tarjeta LPC2468 OEM Board es lenguaje C.

El software necesario para llevar a cabo el desarrollo es el siguiente:

- Editor de texto convencional (o herramienta de desarrollo de C). Se recomienda utilizar Gedit.
- Compilador cruzado de C. Es un compilador de C tradicional, que compila aplicaciones desde un entorno a otro distinto.

La forma más sencilla para el desarrollo de aplicación, si se utiliza la máquina virtual de Embedded Artists, es hacerlo directamente sobre la ubicación /home/user/dev ya que contiene todos los permisos necesarios para el desarrollo. Si se desea utilizar otra ubicación habrá que cambiar la ruta de de ucfront-gcc como corresponda.

Una vez escrito el programa, para compilarlo será necesario utilizar un Makefile. Se recomienda utilizar el que se proporciona a continuación (cambiando <aplicacion> por el nombre de la aplicación que se esté desarrollando).

```
.PHONY: <aplicacion>
all: <aplicacion>
<aplicacion>: <aplicacion>.c
	../uClinux-dist-20051014/tools/ucfront-gcc arm-elf-gcc -Os -
fomit-frame-pointer -fno-common -fno-builtin -Wall -DEMBED -Dlinux -
D__linux__ -Dunix -D__uClinux__ -Wl,-elf2flt="-r" *.c -o <aplicacion>
clean:
	rm -f *~ *.gdb <aplicacion>
```

Finalmente desde un terminal, accedemos a la ubicación donde se encuentre el código fuente y el Makefile, y se ejecutara el comando `make` para realizar la compilación. El resultado de ésta será un fichero binario llamado <aplicacion> (el nombre que se haya seleccionado).

Para instalar el programa desarrollado en la tarjeta, se puede utilizar FTP, NFS o directamente copiándolo en una memoria SD/MMC y luego insertarla en la ranura correspondiente de la tarjeta.

Una vez el programa esté instalado en la tarjeta, para ejecutarlo desde un programa terminal, accedemos a la ubicación de la memoria SD/MMC y escribimos el nombre del binario, en este caso <aplicacion>.

6.1.4. Instalación del driver FTDI USB de la tarjeta LPC2468

Es necesario instalar un driver especial para que funcione chip USB a UART. El driver para cada uno de los sistemas operativos está disponible, como el resto del software en el CD, en el directorio **drivers** y en el subdirectorio correspondiente a cada SO.

En esta sección se va a detallar la instalación y configuración del driver sobre el sistema operativo Windows XP. Para el resto de casos, es el usuario quien debe estudiar este proceso, pero en todos ellos habrá que seguir una secuencia de pasos parecida.

Cuando la tarjeta se conecte al PC a través del cable USB, el equipo preguntará por el driver. Hay que descomprimir el fichero que contiene dicho driver y seleccionar la opción de búsqueda manual para indicar al asistente donde encontrarlo.

Una vez que el driver haya sido instalado correctamente el sistema operativo del PC habrá creado un puerto COM, pero antes de iniciar la comunicación es necesario realizar una configuración previa de dicho puerto. Para ello hay que ir a:

Inicio → Panel de Control → Sistema → Hardware → Administrador de Dispositivos → Puertos (COM & LPT)

Una vez en la ventana del administrador hay que hacer doble clic en el puerto que se ha creado nuevo, para pasar a la ventana de configuración, y fijamos 115200 bits por segundo, 8 bits de datos, sin paridad, 1 bit de parada, y sin control de flujo, tal y como se puede ver en la figura 37.

Algunas aplicaciones, como por ejemplo el programa Philips FLASH Utility (para la descarga por ISP), utilizan puertos del 1 al 5. Por eso si el puerto asignado no está entre esos valores, es necesario acceder a las opciones avanzadas (Avanced) y probar a cambiar el valor del puerto por uno más bajo. Por lo general no deberíamos tener problemas en realizar esta operación.

Finalmente ya estaría todo correctamente instalado. Para probarlo, sólo es necesario conectar la tarjeta con el ordenador mediante USB, abrir un programa terminal y fijar en la conexión los mismos parámetros que se han configurado en el puerto.

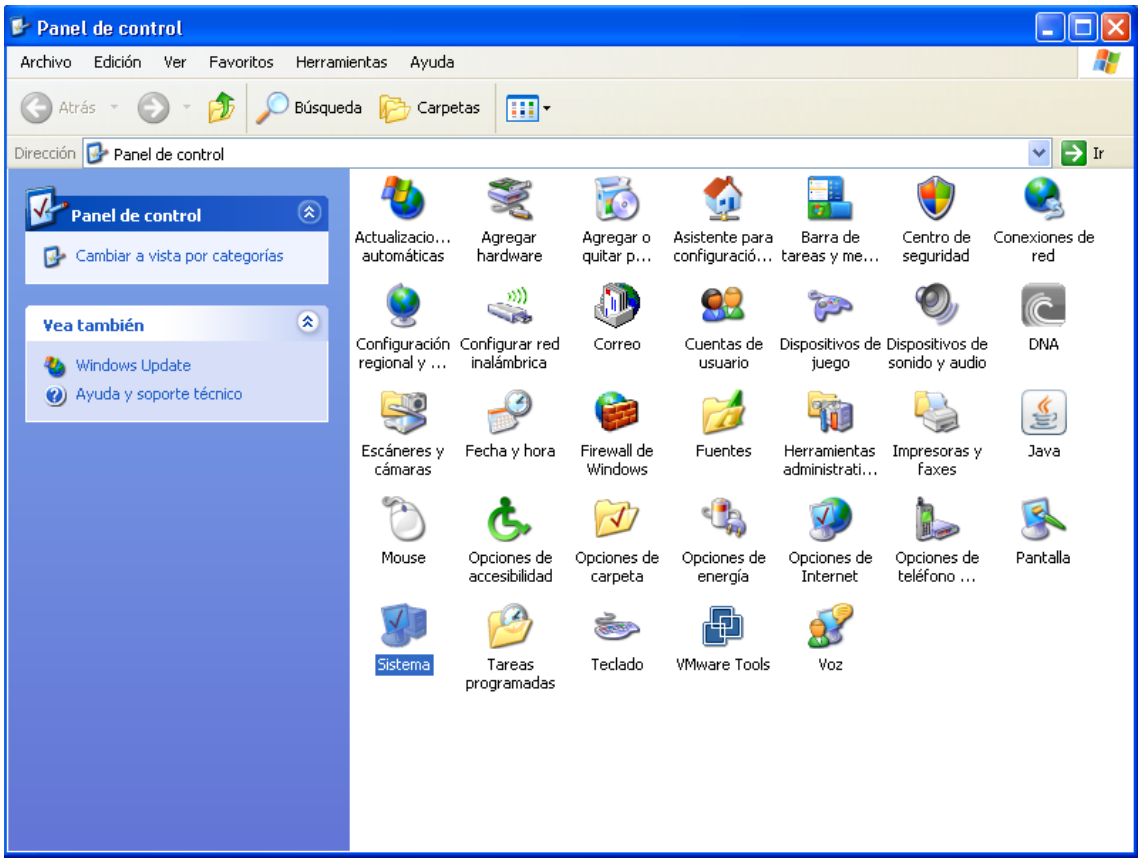


Figura 35. Panel de control de Windows XP

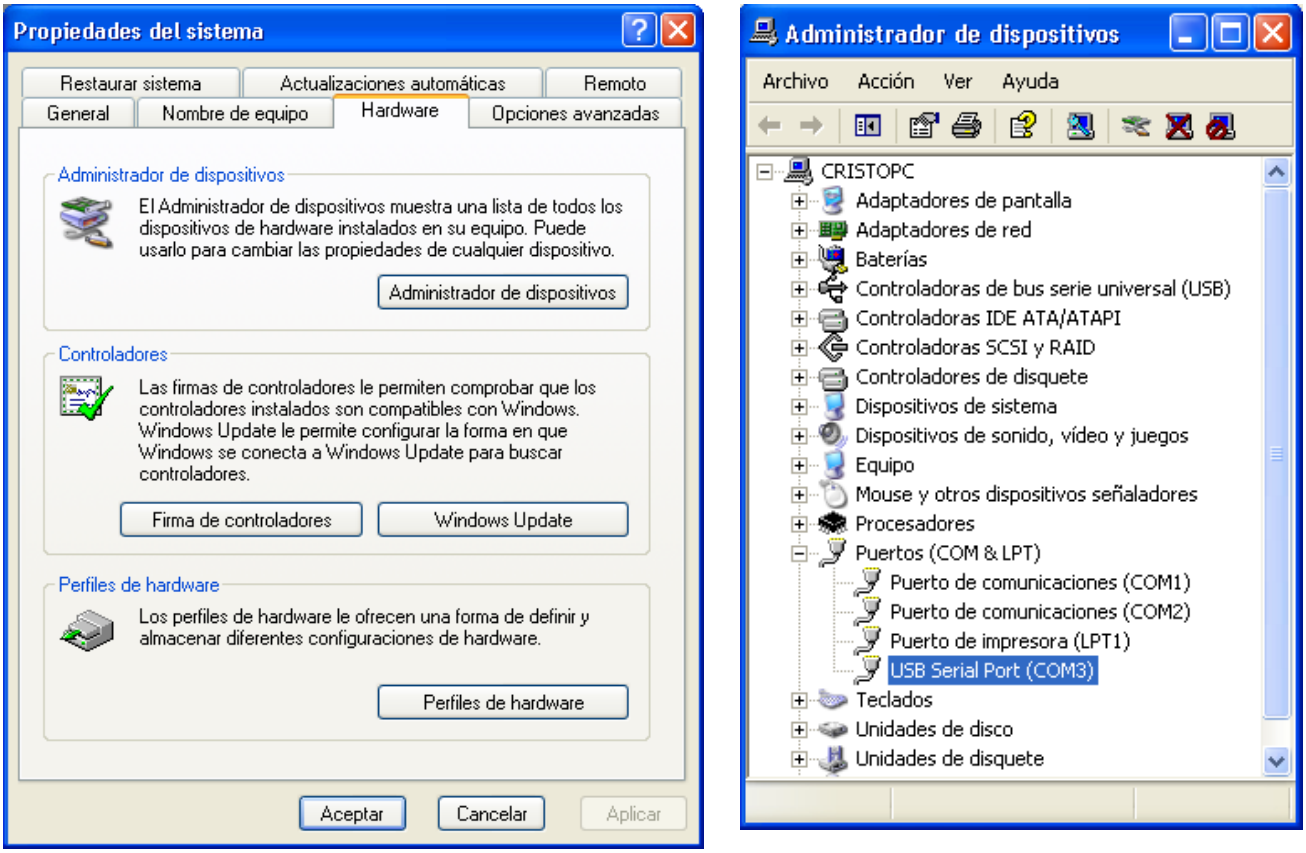


Figura 36. Propiedades de Sistema (dcha) y administrado de dispositivos (izda)

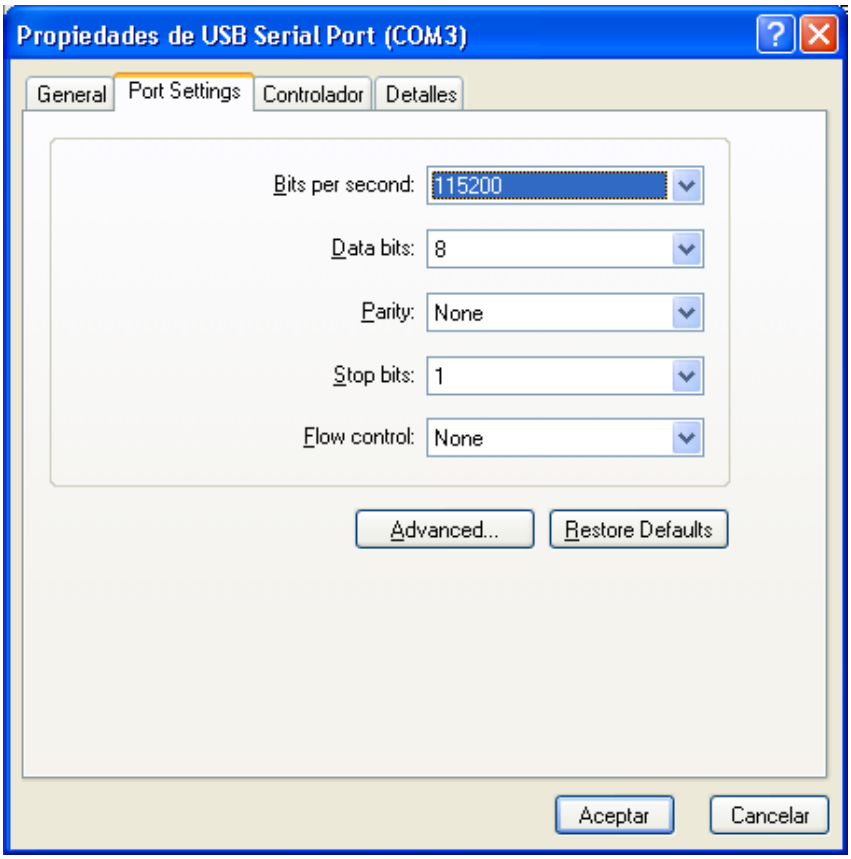


Figura 37. Ventana de propiedades de un puerto COM

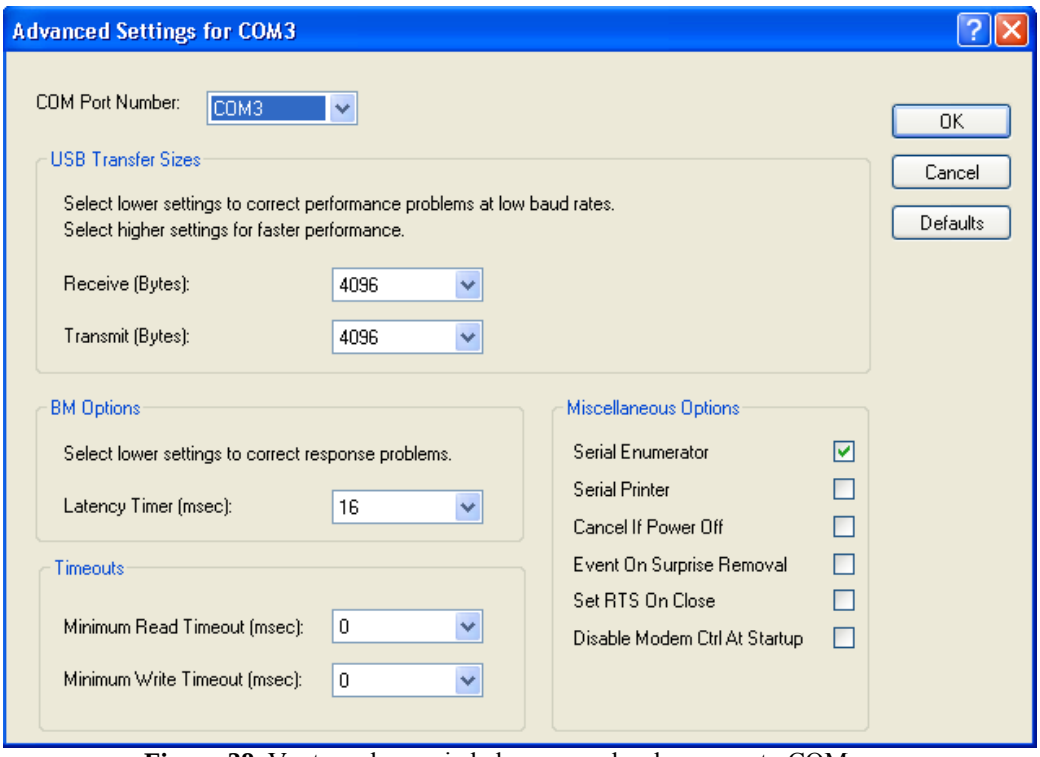


Figura 38. Ventana de propiedades avanzadas de un puerto COM

6.1.5. Conexionado de la tarjeta con el equipo

En primer lugar es necesario comunicar la tarjeta LCP2468 OEM Base Board con el equipo desde el cual se van a descargar las aplicaciones. Los requisitos necesarios para poder establecer la comunicación son:

- Un sistema operativo como Windows, Linux, MacOS, etc...
- Un cliente Telnet, FTP y/o SSH.
- Un programa terminal (Hyperterminal, minicom, etc...).
- Un cable USB (mini-AB a A) para interconectar el equipo con la tarjeta.
- Un cable de ethernet para conectar el equipo con la tarjeta. Tiene que ser cruzado si conectan directamente, o normal si se conectan a través de un dispositivo de red (router, hub, switch, etc...).
- Opcionalmente un transformador para la alimentación, 9-15V DC de 2ª, con conector de 2.1mm y cualquier polaridad.

Existen diversas formas de realizar la interconexión:

- Conectando la tarjeta directamente con el PC a través del puerto USB de ésta, usando el cable mini-AB a A. En este caso la tarjeta puede utilizar directamente la alimentación del puerto USB o utilizar un transformador externo. Es posible tener ambas alimentaciones simultáneamente. La configuración sería la siguiente:

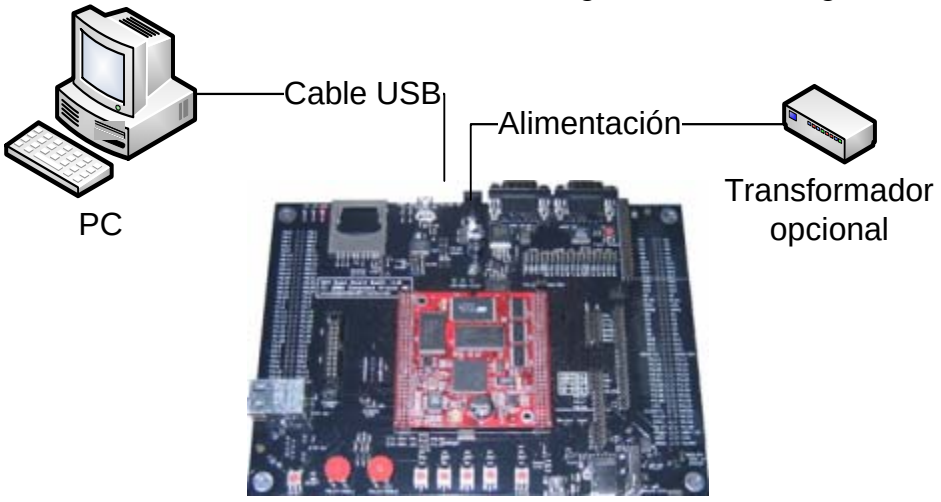


Figura 39. Esquema de alimentación entre el PC y la tarjeta OEM Base Board.

La tarjeta OEM Base Board contiene un chip que convierte de interfaz USB a interfaz RS232 (FT232R a FTDI), que conecta la UART#0 de la tarjeta a un puerto COM virtual del ordenador. Lógicamente es necesario instalar un driver especial para que el PC reconozca la tarjeta (*véase sección 6.1.4, Instalar el driver FTDI USB de la tarjeta LPC2468*).

Existen cuatro jumpers en la tarjeta LPC2468 OEM Base Board relacionados con el canal serie USB, conectados a la UART#0 de la LPC2468. La imagen 40 representa donde se encuentran dichos jumpers. Es muy importante asegurarse de que los jumpers *automatic ISP* estén abiertos. Si no lo están, es posible que el programa terminal haga un reset de la tarjeta y/o habilite el modo ISP por error.

Estos jumpers suelen estar abiertos normalmente, la única excepción es cuando se necesita cargar una nueva versión del autoarranque (u-boot).

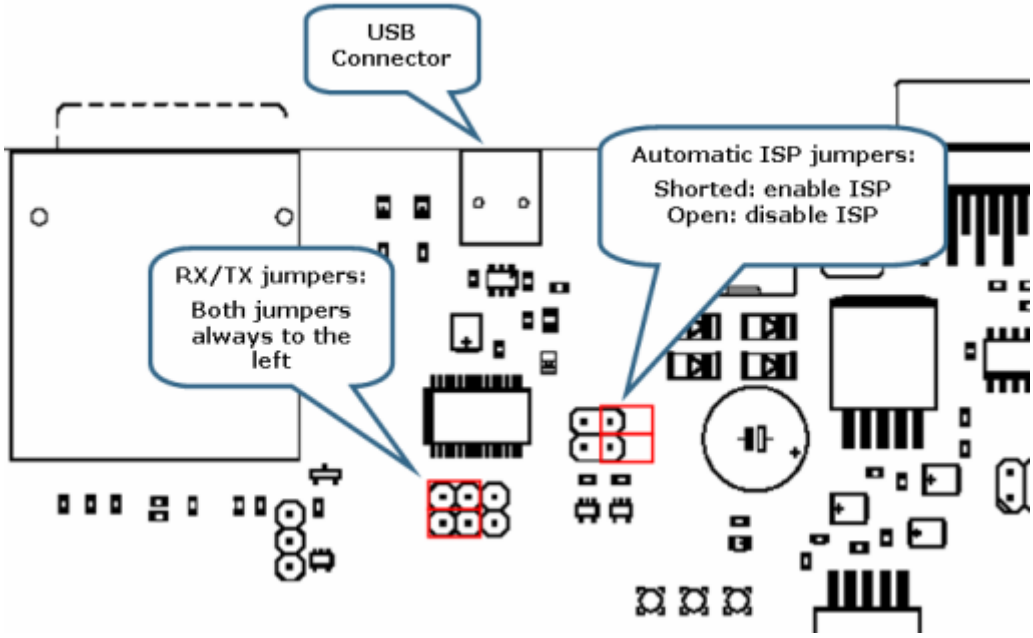


Figura 40. Configuración de los jumpers para el conexionado entre el PC y la tarjeta LPC2468

Los siguientes pasos que hay que seguir para conectar con la interfaz de consola del sistema son los siguientes:

1. Conectar el cable USB entre el PC y la OEM Base Board.
2. Verificar que se activa el LED de alimentación de la OEM Base Board.
3. Buscar el nuevo puerto COM que se ha creado y asegurarse de que está configurado correctamente (*véase sección 6.1.4, Instalación del driver FTDI USB de la tarjeta LPC2468*).
4. Cerciorarse de que los jumpers referentes al **automatic ISP** estan abiertos, y que los jumpers correspondientes a RX/TX estan cerrados, como se indica en la figura 40.
5. Iniciar el programa terminal (Hyperterminal, minicom, ...).
6. Pulsar el botón de reset (en la esquina inferior izquierda de la OEM Base Board)

- Conectando la tarjeta LPC2468 OEM Board a una red LAN a través de la interfaz ethernet. Esta configuración sólo es válida en el caso de tener instalados en la tarjeta el programa de autoarranque y el SO uClinux. La figura 41 muestra como es el esquema de conexionado. Este conexionado se puede realizar de dos formas, conectando directamente la tarjeta con el PC a través de un cable de red cruzado, o conectando ambos equipos con cables de red normales a un dispositivo de red (hub, switch, router,...).

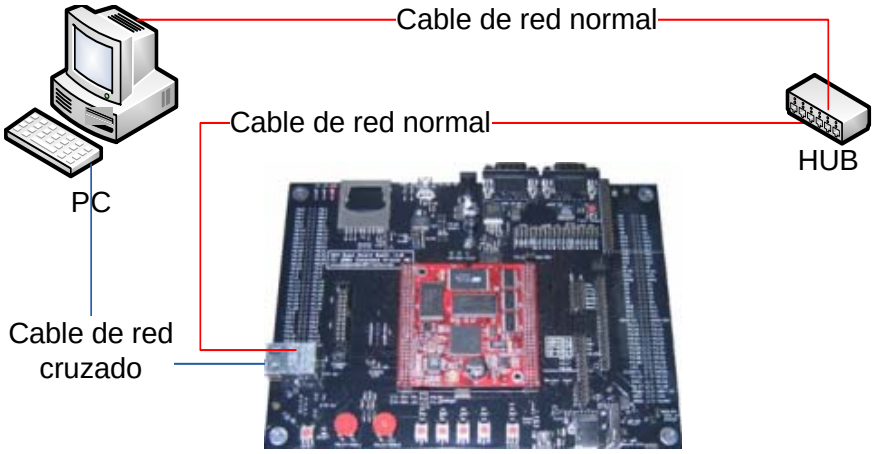


Figura 41. Esquema de conexionado entre el PC y la tarjeta OEM Base Board a través de la red. Las líneas azules muestran la conexión directa, y las rojas la conexión a través de un elemento de red.

El cable de Ethernet debe esta conectado durante el encendido de la tarjeta. La configuración por defecto de la tarjeta LPC2468 OEM Board es:

```
IP address: 192.168.0.100
Netmask: 255.255.255.0
Default gateway: 192.168.0.1
```

Los nodos de una red IP se pueden comunicar directamente con otros que estén en la misma LAN (como por ejemplo en una red Ehthernet) y que estén en la misma subred IP. Esto significa que la tarjeta LPC2468 OEM Board debe estar en la misma subred que el equipo con el que queramos conectarlo, si utilizamos este mecanismo. Generalmente todas las IP's que pertenecen a una red LAN empiezan con el prefijo 192.168.x.x.

En el caso de ser necesario cambiar la IP, se puede optar por cualquiera de estos métodos:

- **Cambiar la IP en el autoarranque –** En este caso la dirección IP está almacenada en la variable de entorno **ipaddress**. Para cambiar esta variable lo único que hay que hacer es arrancar la tarjeta y monitorizarla con un programa terminal. Detener la inicialización pulsando una tecla durante los tres primeros segundos, y escribir la siguiente secuencia de comandos.

```
setenv ipaddress 192.168.0.50
save
```

- **Cambiar la IP en el sistema operativo uClinux –** En este caso la dirección IP se fija cuando se hace la llamada al fichero **rc** en el arranque del SO uClinux. Por defecto esta dirección IP es la 192.268.0.100, pero se puede cambiar fácilmente escribiendo la siguiente secuencia de comandos en la consola del programa terminal una vez finalizado el arranque:

```
ifconfig eth0 down
ifconfig eth0 192.168.2.50 up
```

- **Llamada al servidor DHCP –** En este caso se hace una llamada directamente al servidor DHCP para obtener una dirección IP disponible (que no esté ocupada por ninguna otra máquina de la subred). Para realizar la llamada hay que escribir el comando `dhclient` en la consola del programa terminal una vez finalizado el arranque.

Para poder comprobar que la comunicación se ha establecido correctamente entre la tarjeta OEM Base Board y el PC, hay que seguir los siguiente pasos:

1. Conectar el cable de Ethernet a la tarjeta y encender la alimentación.
2. Asegurarse de que la tarjeta y el PC están en la misma subred. Una forma muy sencilla de probar esto es abriendo una consola en el SO Windows XP del PC, y escribiendo `ping 192.168.0.100`. Cada vez que la interfaz de Ethernet de la tarjeta recibe una trama se enciende un LED verde.

Si todo está bien, entonces la conexión se ha establecido correctamente. En este momento es posible acceder a la tarjeta LPC2468 a traves de los siguientes sistemas:

- **Acceso FTP –** Siempre hay un servidor FTP activo, escuchando en la dirección IP y en el puerto 21.
- **Acceso TELNET –** Siempre hay un servidor TELNET activo, escuchando en la dirección IP y en el puerto 23.
- **Acceso Web/HTTP –** Siempre hay un servidor HTTP activo, escuchando en la dirección IP y en el puerto 80.

Todos estos modos de acceso utilizan un mecanismo de login. En el SO uClinux existen dos cuentas de acceso, una como **root** y otra como **guest** (‘root’ y ‘guest’ respectivamente), ambas con la contraseña **uclinux**.

6.1.6. Acceso a la tarjeta vía FTP

Utilizar un acceso de tipo FTP puede resultar muy útil para actualizar, por ejemplo, los contenidos una SD/MMC cuando el sistema este corriendo, sin necesidad de extraer dicha la SD/MMC, introducirla en un lector, actualizarla, y volverla a colocar en la tarjeta LPC2468 Base Board.

Desde cualquier equipo de la subred se inicializa un cliente FTP. Por ejemplo desde un equipo con Windows XP (otros SO como Linux utilizan los mismo comando u otros muy parecidos) se puede teclear el comando `ftp 192.168.0.100`, siendo la dirección IP la asignada al dispositivo (LPC2468).

Una vez ejecutado el comando hay que seleccionar una cuenta de acceso. Se puede acceder como **root** o como **guest** escribiendo ‘root’ o ‘guest’ respectivamente (para realizar cambios es necesario utilizar ‘root’, ya que ‘guest’ es sólo para consultas), y con el password **uclinux**.

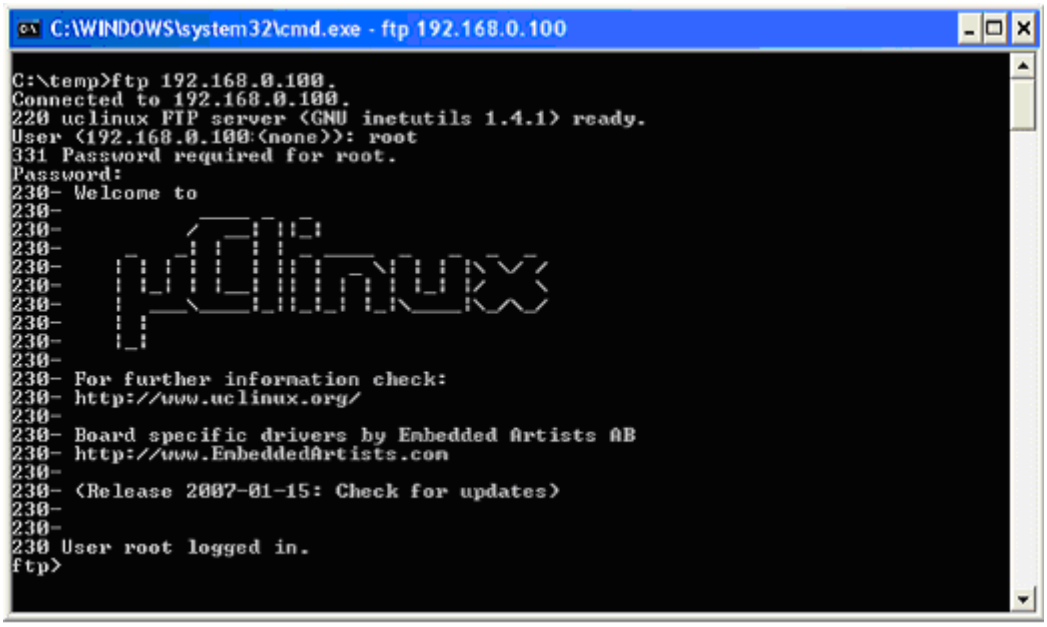


Figura 42. Consola de Windows XP con la llamada FTP.

Una vez que ha sido realizado el login, se pueden ejecutar los siguientes comandos (esta lista muestra los comandos más básicos, aunque existen otros)

- **pwd** – Imprime por pantalla el directorio de trabajo actual.
- **ls** – Imprime el listado de ficheros del directorio de trabajo actual.
- **mkdir** – Crear un nuevo directorio.
- **delete** – Borra un fichero.
- **rmdir** – Borra un directorio
- **put** – Manda un fichero a la tarjeta.
- **get** – Obtiene un fichero de la tarjeta.
- **lcd** – Cambia el directorio actual de trabajo

Entonces para mandar un fichero a la SD/MMC hay que ejecutar los siguientes comandos:

```
ftp>lcd /home/myuser/files
ftp>cd /mnt/mmc
ftp>put afile.txt
```

6.1.7. Acceso a la tarjeta vía TELNET

Realizar la conexión con un cliente TELNET, permite ejecutar comandos y programas, tanto los construidos por los desarrolladores como los integrados en el sistema operativo.

Se puede ejecutar un cliente TELNET desde cualquier equipo que este conectado a la misma subred que la tarjeta. Por ejemplo desde un equipo con Windows XP (otros SO como Linux utilizan los mismos comandos u otros muy parecidos) se puede teclear el comando `telnet 192.168.0.100`, siendo la dirección IP la asignada al dispositivo.

Una vez ejecutado el comando hay que seleccionar una cuenta de acceso. Se puede acceder como **root** o como **guest** escribiendo 'root' o 'guest' respectivamente (para realizar cambios es necesario utilizar 'root', ya que 'guest' es sólo para consultas), y con el password **uclinux**.

Si se utiliza la ventana de comandos de DOS, el resultado será parecido a la figura 43.

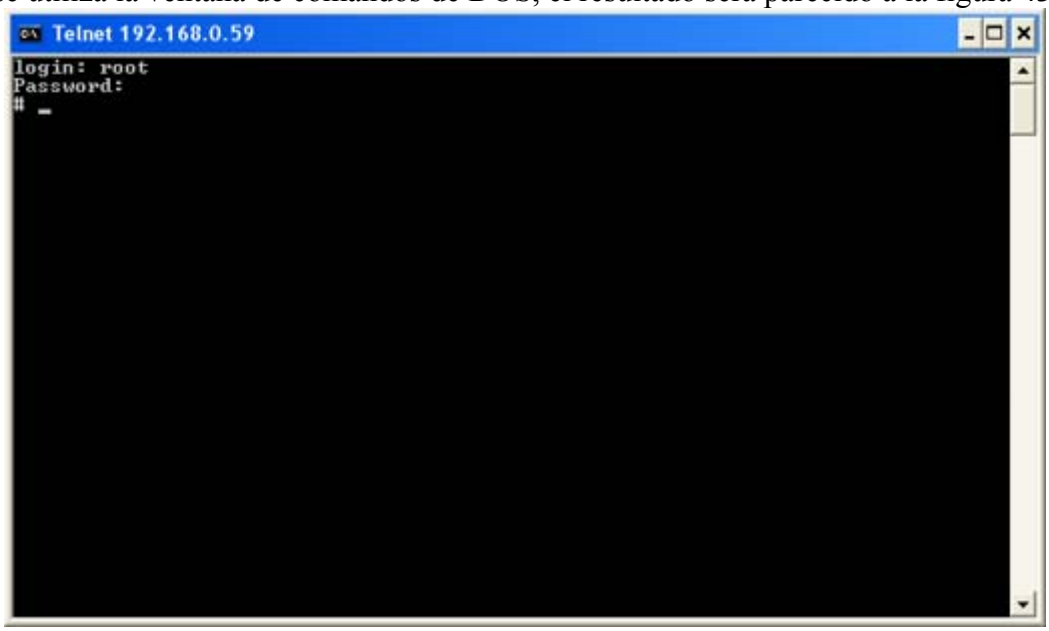


Figura 43. Consola de Windows XP con la llamada TELNET.

Una vez iniciada la sesión, se puede tener acceso a todos los comandos telnet, como por ejemplo:

- **ls** – Imprime el listado de ficheros del directorio de trabajo actual.
- **ping** – Comprueba si un dispositivo esta disponible a través de la red.

- **cat** – Muestra el contenido de un fichero

Para desconectarse del cliente TELNET sólo hace falta escribir el comando `exit`.

Existen otras alternativas para conectarse en modo TELNET, como por ejemplo usando **Hyperteminal**, o **PuTTY**.

6.1.8. Acceso a la tarjeta vía Web/HTTP

La tarjeta LPC2468 Board viene preconfigurada con un servidor web. Si hay una SD/MMC insertada y montada, y tiene el directorio **www**, entonces ese directorio será la raíz del servidor web. Sin embargo, si no hay ninguna tarjeta de memoria insertada, la tarjeta LPC2468 devolverá una página web con el listado del árbol de directorios de ésta (véase figura 44).

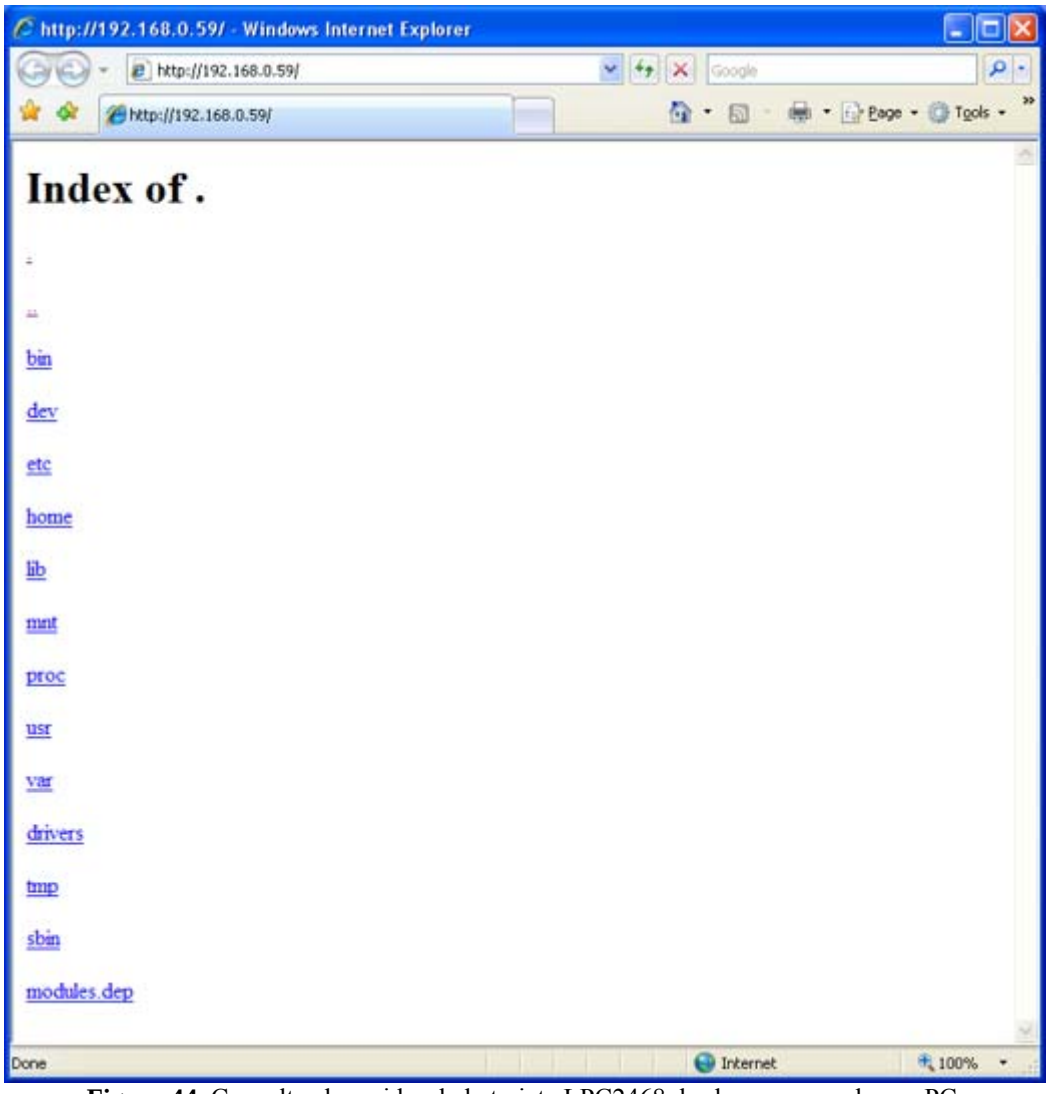


Figura 44. Consulta al servidor de la tarjeta LPC2468 desde un navegador un PC.

6.1.9. Instalación del software de autoarranque (uboot)

En esta sección se explica como instalar el software de arranque en la tarjeta LPC2468.

Este software es un pequeño programa en formato .hex, que se encarga de hacer las llamadas a los distintos componentes hardware del sistema para iniciar la tarjeta.

Se recomienda utilizar el binario ya compilado, aunque es posible recopilarlo para añadir o quitarle alguna funcionalidad.

Hay que tener en cuenta, tanto en la compilación como en el uso del fichero ya compilado, las características de la tarjeta para conseguir que el fichero de arranque (uboot.hex) funcione correctamente. En este caso concreto los parámetros a tener en cuenta son:

- Bus de datos 16 bits.
- Cristal de frecuencia 48MHz o 72MHz.

Por ello el binario que debemos cargar es

u-boot_v1.1.6_lpc2468_16bit_48MHz.hex

que se encuentra en la ubicación **SDK/Software**.
Antes de comenzar necesario conocer cual es el puerto al que va conectada la tarjeta. Esta información se puede obtener consultando

Panel Control → Sistema → Hardware → Administrador dispositivos → Puertos(COM&LPT)

Para poder descargar este programa en la tarjeta hay que conectar los jumpers P2.10 y RESET de la parte superior central de la tarjeta como indica la figura 45.

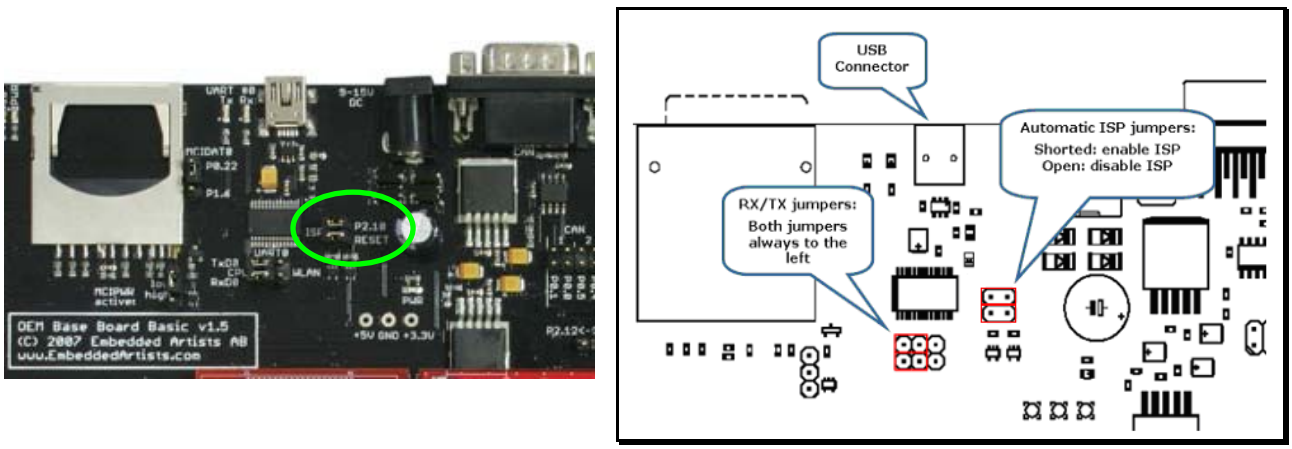


Figura 45. Configuración de jumpers para descargar uboot en la tarjeta LPC2468.

Se necesita utilizar un software adicional para descargar los ficheros. En el desarrollo de este sistema se han tenido en cuenta dos alternativas, aunque existen muchas otras igualmente validas.

Flash Magic	Es sencillo de instalar (sólo es necesario hacer doble click y seguir los pasos). Proporciona una interfaz amigable y de fácil uso.
lpc21isp.exe	No necesita instalación. Se ejecuta desde consola introduciéndole los parámetros deseados.

- Para utilizar **Flash Magic**, tras iniciar el programa, se establece la configuración de la figura 46, y en el campo **Hex File** se selecciona el fichero a descargar.

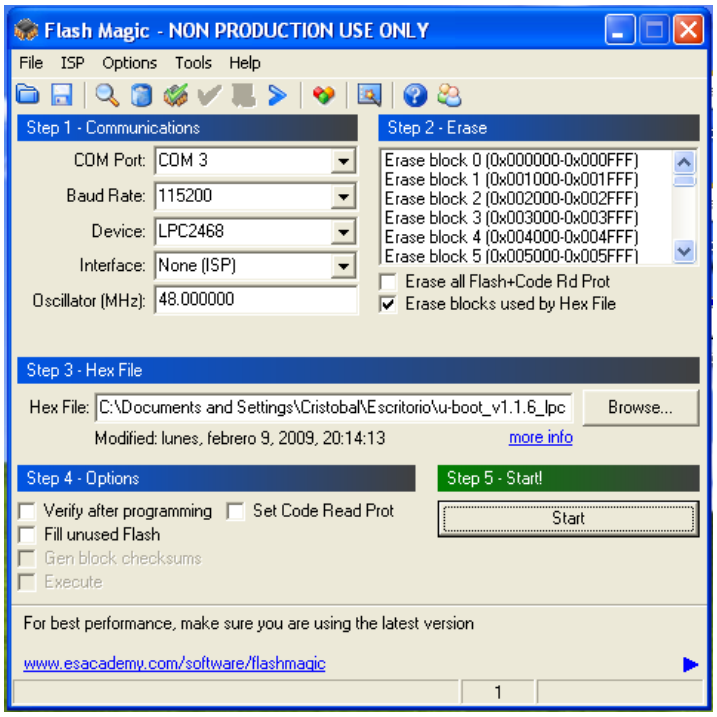


Figura 46. Configuración de Flash Magic para descargar programas

- Para utilizar lpc21isp, se inicia una sesión de consola en el sistema operativo que corresponda (en este caso Windows XP, aunque sería similar en otros), y se pasan por parámetro en la llamada la configuración.

```
lpc21isp u-boot_v1.1.6_lpc2468_16bit_48MHz.hex com3 115200 48000
```

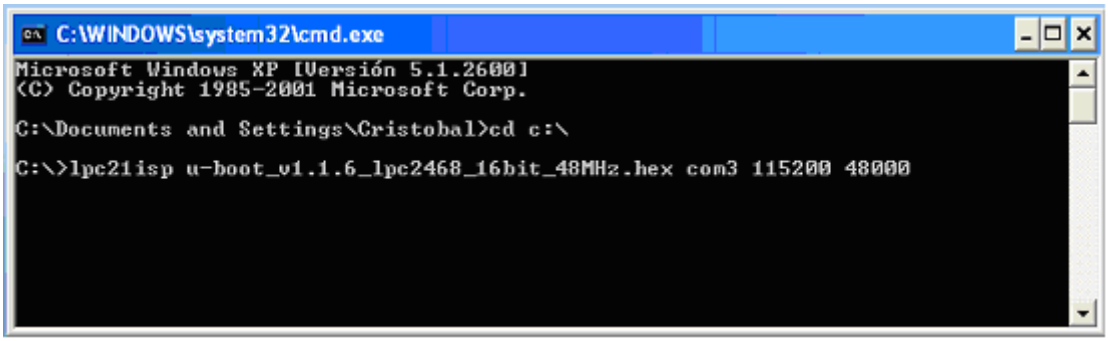


Figura 47. Configuración de lpc21isp.exe para descargar programas

6.1.10. Instalación sistema operativo uClinux en la tarjeta LPC2468

En esta sección se detalla como instalar el sistema operativo uClinux.

Este sistema operativo está basado en una distribución Linux, y está formado por dos ficheros.

uLinux.bin – Es la imagen del sistema operativo uClinux.
romfs.img – Es el sistema de ficheros inicial.

Tabla 6. Ficheros disponibles para el sistema operativo (uClinux)

Estos ficheros se pueden obtener directamente de la ubicación **SDK/Software**, aunque existe la opción de recompilarlos utilizando el entorno de desarrollo **uCLinux** (véase sección 6.1.2, Instalación o actualización del entorno de desarrollo uClinux). Usar esta última opción, permite modificar la configuración de uClinux para adaptarlo a las necesidades concretas de cada diseño.

Para cargar el sistema operativo en la tarjeta, puede utilizarse TFTP, una memoria SD/MMC o carga directa sobre la memoria NAND FLASH. El primer metodo (TFTP) es muy util a la hora de desarrollar, ya que permite modificar el sistema operativo de forma rapida, pero no permanente. Si se desea mantener permanentemente instalado el SO, se recomienda usar la SD/MMC o la copia directa en la NAND FLASH.

6.1.10.1. Carga del sistema operativo a través de TFTP

Cargar el sistema operativo utilizando TFTP es la forma más sencilla y rápida de hacerlo. Este método de carga se utiliza típicamente durante el desarrollo de aplicaciones relacionadas con el núcleo y con su funcionalidad (por ejemplo en el desarrollo de drivers específicos).

No copia las imágenes en la tarjeta LPC2468, sino que las carga directamente en la memoria volátil, por lo que en cada reinicio, se eliminan y hay que volver a cargarlas.

Notar que es necesario tener instalado un servidor TFTP en el sistema de desarrollo (en el PC) y que los ficheros **uLinux.bin** y **romfs.img** tienen que estar ubicados en un directorio accesible, para poder descargarlos. Se recomienda establecer la raíz del servidor TFTP al directorio de las imágenes directamente, para evitar tener que mover los ficheros de un directorio a otro.

También es importante saber que la dirección IP tanto de uClinux como de servidor TFTP se deben fijar correctamente, así como las mascaras de red. Para fijar y guardar estos parámetros hay que escribir los siguientes comandos:

```
setenv ipaddr 192.168.1.100
setenv serverip 192.168.1.3
setenv netmask 255.255.255.0
saveenv
```

Los comandos para la descarga del sistema operativo en la tarjeta serían

```
tftpboot a0008000 linux.bin
tftpboot a1800000 romfs.bin
go a0008000
```

6.1.10.2. Carga del sistema operativo a través de una memoria SD/MMC

Otra forma rápida de transferir las imágenes, es a través de una tarjeta de memoria SD/MMC. Sólo hay que copiar las imágenes (**uLinux.bin** y **romfs.bin**) a la memoria e insertarla en la ranura de la tarjeta OEM Base Board.

Con este método, al igual que con el anterior, las imágenes no se almacenan en la tarjeta LPC2468. Por el contrario se recuperan cada vez de la tarjeta SD/MMC y se cargan directamente en la memoria volátil, por lo que en cada reinicio se borran y hay que volver a cargarlas.

Realmente esto no supone un inconveniente, puesto que la memoria SD/MMC al estar instalada dentro de la tarjeta, forma parte de la memoria de almacenamiento, aunque debe permanecer instalada en cada reinicio. Si no se dispone de una de estas memorias, entonces no es posible cargar el sistema operativo de esta forma.

A continuación hay que seguir los siguientes pasos:

1. Copiar las imágenes en el directorio raíz de la tarjeta de memoria SD/MMC
2. Reiniciar las variables de entorno.

```
protect off 7c000 0x7cfff
erase 7c000 7cfff
```

3. Cambiar la instrucción de autoarranque para que cuando se reinicie arranque desde la tarjeta SD/MMC.

```
setenv bootcmd run mmc_boot
```

4. Guardar las variables de entorno que han sido modificadas conservarlas permanentemente.

```
saveenv
```

6.1.10.3. Carga del sistema operativo directamente la memoria NAND FLASH

Con este método se consiguen cargar de forma permanente las imágenes del sistema operativo en la memoria de almacenamiento de la tarjeta LPC2468.

Es un método más laborioso, pero permite reiniciar la tarjeta cuantas veces se desee sin necesidad de tener que cargar manualmente las imágenes.

Para poder cargar las imágenes la primera vez, es necesario pasarlas a la tarjeta por algún mecanismo de los explicados anteriormente. Los pasos que hay que seguir son los siguientes:

1. Copiar los ficheros **uLinux.bin** y **romfs.bin** a la tarjeta de memoria SD/MMC, preferiblemente en el directorio raíz.
2. Introducir la tarjeta SD en la ranura especifica de la LPC2468.
3. Reiniciar la tarjeta LPC2468 y para el uboot antes de que pasen 3 segundos, pulsando una tecla.
4. Reiniciamos las variables de entorno.

```
protect off 7c000 0x7cfff
erase 7c000 7cfff
```

5. Borramos la nand, para poder introducir allí el código del SO y del sistema de archivos.

```
nand erase
```

6. Arrancamos la tarjeta SD.

```
mmc
```

7. Copiamos el SO en la RAM para posteriormente pasarlo a la NAND FLASH.

```
fatload mmc 0 a1500000 uLinux.bin
```

8. Pasamos el SO a la NAND FLASH para poder tenerlo permanetemente aunque se apague la tarjeta LPC2468.

```
nand write a1500000 0 300000
```

9. Copiamos el sistema de ficheros en la RAM para posteriormente pasarlo a la NAND FLASH.

```
fatload mmc 0 a1800000 romfs.img
```

10. Pasamos el SO a la NAND FLASH para poder tenerlo permanetemente aunque se apague la tarjeta LPC2468.

```
nand write a1800000 300000 400000
```

11. Cambiamos la instrucción de autoarranque para que cuando se reinicie arranque desde las NAND FLASH (es ***una sola línea***).

NOTA: Hay que tener en cuenta que los valores de las dimensiones de memoria asignados están dimensionados para las imágenes que ya han sido generadas. En caso de compilar imágenes nuevas, sería necesario recalcular dichas dimensiones para no solapar zonas de memoria

```
setenv 'nand_boot nand read a1500000 0 300000; nand read a1800000
300000 400000; bootm a1500000'
```

12. Guardamos las variables de entorno que hemos modificado para tenerlas permanentemente.

```
saveenv
```

6.1.11. Instalación de la aplicación de control de acceso.

En esta sección se detalla como llevar a cabo la instalación del software desarrollado para la tarjeta LPC2468 una vez esté instalado el sistema operativo uClinux.

Para la instalación de la aplicación hay que seguir los siguientes pasos:

1. Comprobar los jumpers para que la configuración sea la correcta (*véase sección 6.1.5, Conexionado de la tarjeta con el equipo*).
2. Copiar los ficheros **usbaccess** y **userrc** en una tarjeta de memoria SD/MMC.
3. Introducir la tarjeta en la ranura correspondiente de la tarjeta OEM Base Board.
4. Conectar la pantalla QVGA en el puerto de expansión correspondiente.
5. Conectar la fuente de alimentación a la tarjeta OEM Base Board.

Después del último paso, la tarjeta iniciará automáticamente el sistema operativo y lanzará también, de forma automática, la aplicación.

6.1.12. Configuración de los jumpers

En esta sección se detallan las distintas configuraciones de pines que deben aplicarse a la tarjeta OEM Base Board para cada uno de los funcionamientos que se desean conseguir.

6.1.12.1. Conexión de la tarjeta con el PC a través de USB

Para llevar a cabo la interconexión de la tarjeta con el PC es necesario configurar los jumpers de la siguiente manera:

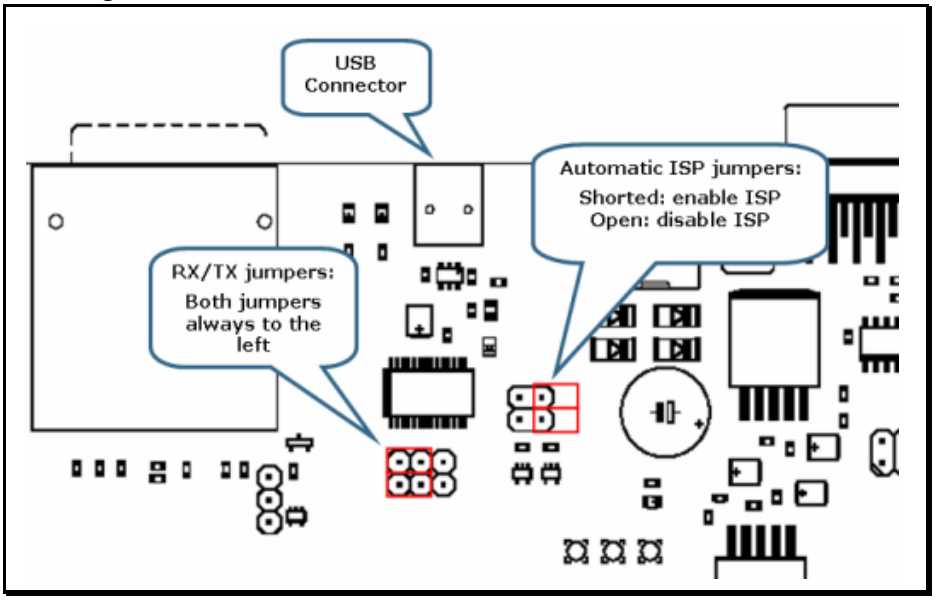


Figura 48. Jumpers para conexión de la tarjeta a través del puerto USB.

6.1.12.2. Descarga directa de programas desde el PC a la tarjeta a través de USB

Para llevar a cabo la interconexión de la tarjeta con el PC con el objetivo de descargar programas utilizando Flash Magic u otra aplicación similar, es necesario configurar los jumpers de la siguiente manera:

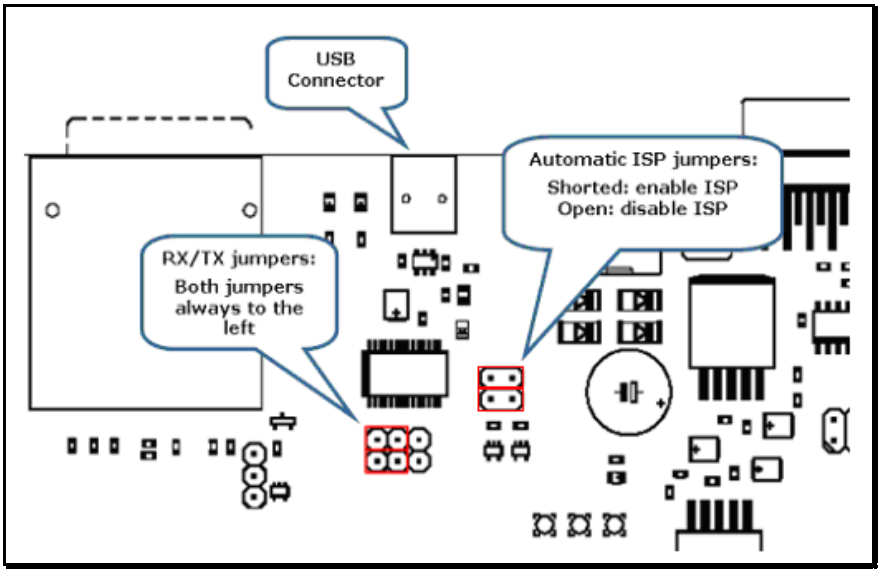


Figura 49. Jumpers para la descarga directa de programas a la tarjeta.

6.1.12.3. Configuración del puerto USB como host

Para que el puerto USB se comporte como host, hay que colocar el jumper P1.30 (esquina inferior derecha de la tarjeta) en la posición **normal**, tal y como se indica en la figura 50.

Figura 50. Jumper para configurar el puerto USB como host.

El driver que permite la lectura y escritura en el módulo de USB, está instalado directamente en el núcleo del sistema operativo, por lo que no es necesario cargar ningún módulo para iniciar el proceso.

Para poder acceder al dispositivo USB es necesario montar previamente la unidad. Para ello hay que ejecutar el siguiente comando

```
mount -t vfat /dev/sda1 /mnt/usbmem
```

Para extraer el dispositivo de forma segura hay que ejecutar el siguiente comando

```
umount /mnt/usbmem
```

6.1.12.4. Configuración de la pantalla táctil

El controlador del display QVGA integrado en la tarjeta, puede ser configurado para diferentes opciones de interfaz según la configuración de los jumpers. La OEM Base Board tiene un conjunto de jumpers con la leyenda impresa al lado para poderlos configurar correctamente.

Para seleccionar la interfaz paralela de 16 bits se tienen que configurar los jumpers según la tabla 7 (los 6 jumpers de arriba se encuentran cerca del pin 1 del módulo del display y los 7 jumpers de abajo se encuentran al final del conector de 2x20 del módulo del display).

Jumper	Posición
PSX/CFG1	High
DTX1/CFG2	Low
DTX2/CFG3	Low
BWS0/CFG4	High
BWS1	does not matter
BWS2	does not matter
CS	left position (to CS2)
RST	left position (to RST)
RS	left position (to A1)
WR/RW	left position (to WE)
P0.15	leave open
P0.17	leave open
P0.18	leave open

Tabla 7. Configuración de los jumpers para la pantalla táctil.

Para que la pantalla funcione con la interfaz táctil, es necesario conectar los jumpers P0.15, P0.17 y P0.18.

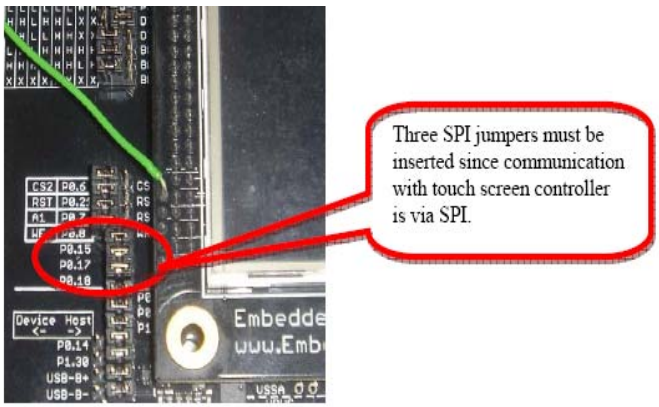


Figura 51. Jumper para la comunicación con la pantalla táctil.

6.2. Configuración de la aplicación software en equipos de laboratorios

El software de control de los equipos del laboratorio va instalado directamente sobre dichos equipos.

A continuación se muestran los diagramas de flujo para cada tipo de instalación.

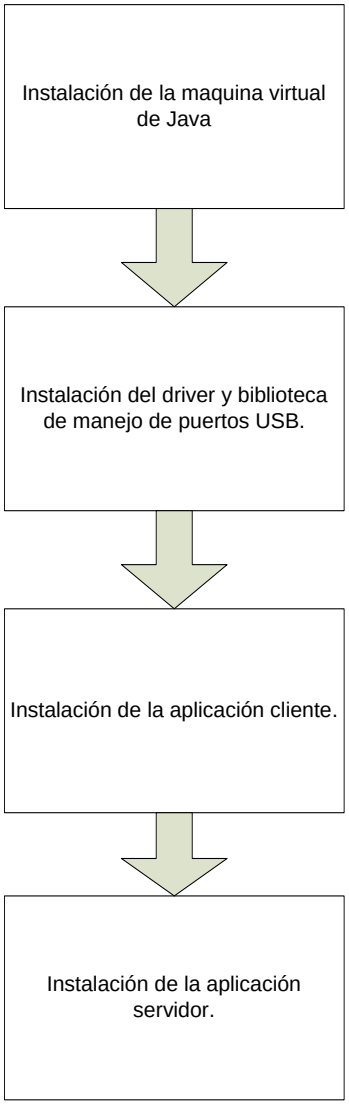


Figura 52. Secuencia de pasos para la instalación de la aplicación de control software.

En los siguientes apartados se desarrolla en detalle cada uno de los pasos del diagrama mostrado anteriormente.

6.2.1. Instalación de la máquina virtual de Java.

Toda la aplicación ha sido desarrollada en lenguaje Java. Por lo tanto es necesario disponer de la máquina virtual para poder ejecutarla. En esta sección se detallan los

pasos que hay que seguir para llevar a cabo dicha instalación.

La versión de la máquina virtual que se necesita es la 1.4.1.03 o superior. Hay que tener en cuenta que versiones superiores a la especificada no son compatibles con algunas de las funciones del módulo de control USB. Aún así, el funcionamiento de la aplicación no se ve alterado.

Los instalables del SDK y del JRE, de la versión 1.4.1.03 de Java, están disponibles en el CD-ROM del proyecto, en la ubicación **Módulo_Software/MVJ/ j2sdk-1_4_1_03-windows-i586.exe** y **Módulo_Software/MVJ/j2re-1_4_1_03-windows-i586-i.exe** respectivamente. Para el funcionamiento de la aplicación sólo es necesario es JRE. La opción de instalación del SDK es para facilitar el entorno de desarrollo a los usuarios de esa máquina.

Además, esta versión y las posteriores están disponibles en la página de Sun Microsystems (www.sun.com).

Para instalar la máquina virtual, junto con su entorno de ejecución y/o desarrollo, sólo es necesario hacer doble clic sobre el ejecutable del CD o el descargado y seguir las sencillas instrucciones del instalador.

6.2.2. Instalación del driver y biblioteca de manejo de puertos USB.

En esta sección se describe como instalar el API Java USB para Windows. Para la instalación de todos los componentes hay que seguir la siguiente secuencia de pasos:

1. Copiar el fichero jusb.dll de la carpeta **\Módulo_Software\USB\JusbDll** a la carpeta **\system32** del directorio de Windows.
2. Registrar el driver **JUSB** en el registro de Windows. Para ello, hay que hacer doble click sobre el fichero llamado **jusb.reg** en la ubicación **\Módulo_Software\USB\JusbDriver**. Aparecerá una ventana preguntando si se desea añadir la información de **jusb.reg** al registro. Después de pulsar en el botón aceptar, se mostrará otra ventana con la confirmación y la información será añadida al registro. Este proceso para registrar el driver **jUSB** sólo tiene que realizarse una vez.
3. Después de haber registrado el driver jUSB, es necesario copiar el fichero jusb.sys (el driver) ubicado en la carpeta **\Módulo_Software\USB\JusbDriver**, a la carpeta **\system32\drivers** del directorio de Windows.

6.2.3. Instalación de la aplicación cliente.

En esta sección se describe como instalar los ficheros de la aplicación en los equipos que utilizarán los usuarios.

Para ello hay que seguir los siguientes pasos:

1. Copiar todo el contenido del directorio **\Módulo_Software\Aplicacion\Cliente** a la carpeta **\Archivos de programa\calus**.
2. Crear un acceso directo del fichero calus.jar y copiarlo en **C:\Documents and Settings\User\Menú Inicio\Programas\Inicio** (donde User debe coincidir con el nombre del equipo).

De esta forma la aplicación se ejecutará automáticamente al iniciar el equipo.

6.2.4. Instalación de la aplicación servidor.

En esta sección se describe como instalar los ficheros de la aplicación servidor en el equipo del administrador.

Para ello hay que seguir los siguientes pasos:

1. Copiar todo el contenido del directorio **\Módulo_Software\Aplicacion\Servidor** a la carpeta **\Archivos de programa\calusServer**.
2. Crear un acceso directo del fichero calus_server.jar y copiarlo en **C:\Documents and Settings\User\Menú Inicio\Programas\Inicio**.

De esta forma la aplicación se ejecutará automáticamente al iniciar el equipo.

7. Análisis del sistema

En esta sección se exponen las ventajas e inconvenientes que presenta el sistema, así como unas estadísticas de uso por parte de los usuarios de la universidad.

Las ventajas que tiene este sistema

- **Facilidad de utilización –** Los usuarios sólo tienen que introducir una memoria USB en la ranura correspondiente y teclear una clave para acceder a ambos sistemas.
- **Alta seguridad –** El sistema se basa en el uso de una doble llave, la memoria USB y la clave del usuario, lo que hacer que sea muy complicado realizar técnicas de:
  - **Suplantación** – Un usuario no puede utilizar la llave USB de otro, puesto que también necesitaría la contraseña de acceso.
  - **Acceso no autorizado –** Sin la llave y la contraseña simultáneamente no se puede tener acceso a ninguno de los dos sistemas.
  - **Manipulación de los datos generados –** Los ficheros de datos también están cifrados, por lo que ni siquiera el propio usuario dueño de los datos puede modificarlos.
- **Independiente –** El sistema no depende de una base de datos de contraseñas externa, lo que permite seguir utilizándolo aún cuando la red de comunicaciones se haya caído. Esta independencia aumenta la calidad del sistema.
- **Sistema multiplataforma –** Otra de las ventajas del sistemas de acceso a los equipos, es que funciona sobre cualquier sistema operativo, lo que amplia el rango de implantación del mismo.
- **Bajo coste –** En la actualidad los dispositivos USB de almacenamiento masivo se producen a gran escala, lo que ha permitido una reducción de su precio en los últimos años. Esto hace que proporcionar a los alumnos y al resto del personal de la universidad un dispositivo de este tipo para el acceso sea poco costoso. Todo esto proporciona una ventaja frente a otras tecnologías como las tarjetas inteligentes o de RFID, ya que al producirse menor número de unidades son más caras.
- **Rapidez y facilidad para generar repuesto –** En el supuesto caso de que un usuario del sistema perdiera su llave o esta quedará inutilizada por cualquier motivo (corrupción de datos, formateo voluntario o involuntario, destrucción del dispositivo, ...) generar una nueva llave sería cuestión de unos pocos minutos. Este hecho proporciona ventaja frente a otro tipo de sistemas de autenticación basados en dispositivos portátiles como las tarjetas inteligentes o las de RFID, ya que generar un nuevo dispositivo lleva varios días o incluso semanas.

Las desventajas del sistema son:

- **Olvido de contraseña** – Al igual que cualquier sistema basado en una contraseña que el usuario debe memorizar, se puede olvidar. No existe ninguna forma implementada en esta versión para recuperar la contraseña, por lo que el usuario tendrá que generar una nueva, y por lo tanto volver a crear el fichero llave.
- **Perdida de la memoria USB con la llave –** No genera ningún fallo de seguridad, puesto que nadie puede usar una llave USB sin la contraseña del usuario. Los datos con las estadísticas de uso del usuario (si no existiera una copia de seguridad por parte del usuario) están replicados en las bases de datos de la universidad. Para obtener una nueva llave, el usuario sólo tiene que acceder a su cuenta de Plataforma Docente (Aula Global 2) y hacer clic en el enlace que llamado **Generar fichero llave USB**. Una vez descargado el fichero generado, sólo hay que copiarlo nuevamente en una memoria USB nueva.

En las tablas 7, 8 y 9, y en la figura 53 se presentan las estadísticas de uso del sistema por parte de los usuarios.

Nº de usuarios que han usado el sistema	10
Nº de accesos validos con llave y clave correcta	10
Nº de accesos inválidos con llave y clave correcta	0

Tabla 8. Número de usuarios y de accesos validos y fallidos

Parámetro medido	Valoración media
Facilidad de uso	8.4
Tiempo de acceso	7.6
Comodidad de uso	8.1
Seguridad	9.6

*Todas las valoraciones son entre 0 y 10, siendo 0 el peor resultado y 10 el mejor

Tabla 9. Media de los resultados totales

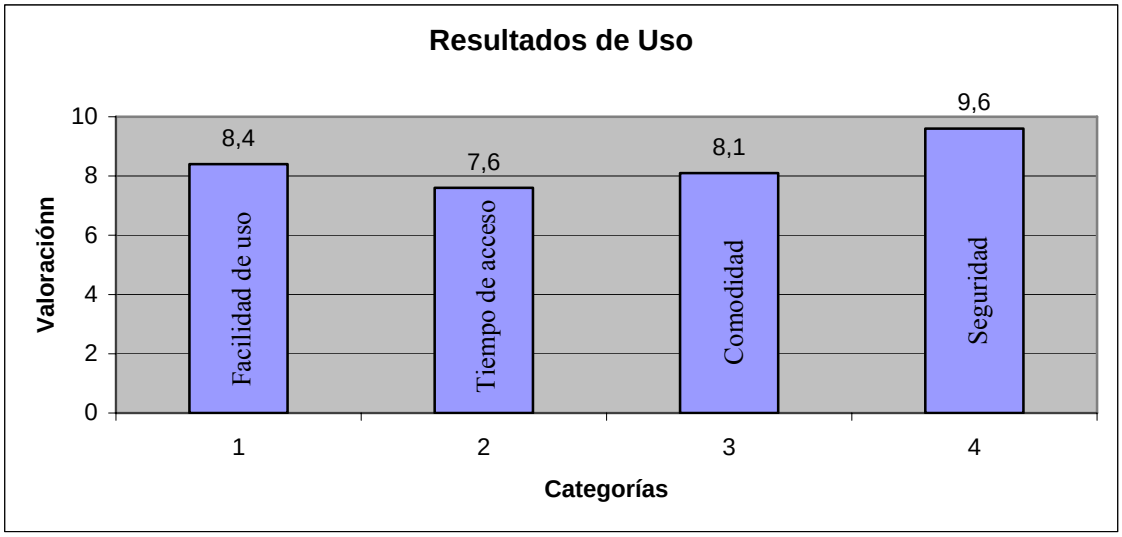


Figura 53. Representación de los resultados de uso

Nombre del usuario	Facilidad de uso	Tiempo de acceso	Comodidad de uso	Seguridad
100047297	9	8	8	9
100074246	9	8	9	10
A0001	8	7	7	10
A002	8	8	8	10
A003	8	7	8	10
A004	8	7	8	10
A005	9	8	9	9
A006	8	8	8	9
A007	9	8	9	10
A0008	8	7	7	9

*Todas las valoraciones son entre 0 y 10, siendo 0 el peor resultado y 10 el mejor

Tabla 10. Resultados parciales de cada usuario

8. Conclusiones

En este apartado se exponen todas las conclusiones que se han obtenido con la realización de este proyecto.

Desde el punto de vista del desarrollo del proyecto, se puede decir que se han conseguido todos los objetivos esperados de CALUS (*véase sección 1.1, Ámbito general del problema y Objetivos*):

- Se ha conseguido un control de acceso a los laboratorios y aulas informáticas de la Universidad Carlos III de Madrid.
- Se ha conseguido un control de acceso en los equipos de los laboratorios docentes y aulas informáticas de la Universidad Carlos III de Madrid.
- Se ha conseguido crear un registro de actividad de cada uno de los usuarios para su posterior evaluación.
- Se ha creado una aplicación de gestión que facilita enormemente las labores administrativas, además de ofrecer un conjunto completo de estadísticas de accesos.

  Por una parte, beneficia a los alumnos porque les permite ser evaluados de forma objetiva, y por otro a los profesores porque facilita los métodos de evaluación continua y trabajo personal.
- Se ha establecido la plataforma básica que permitirá incluir actividades de evaluación remota, autoevaluación, entrega de cuestionarios y/o prácticas, foros de debate *on-line*, teleaprendizaje, etc.

Desde el punto de vista del autor del presente proyecto, se han obtenido las siguientes conclusiones:

- La realización de este proyecto ha servido para aplicar muchos de los conocimientos adquiridos durante la carrera y, además, ha sido una excelente oportunidad para comprobar su utilidad.
- Se ha adquirido experiencia en el uso de herramientas de desarrollo como Eclipse y uClinux.
- Se ha adquirido experiencia en el desarrollo e implementación tanto de aplicaciones como de drivers integrados en sistemas hardware específico, como lo es la tarjeta LPC2468.
- Se ha adquirido habilidad en el uso, configuración e instalación de sistemas operativos, drivers y aplicaciones en sistemas hardware específico como lo es la tarjeta LPC2468.

9. Líneas de trabajo futuro

En este punto se proponen ampliaciones y mejoras al sistema CALUS que pueden proporcionar nuevos beneficios y permitirán sacar mayor provecho de él.

- Implementar una aplicación web que permita que cada usuario pueda consultar de forma remota las estadísticas de uso de cada una de las herramientas software de ayuda al diseño, o los programas informáticos de propósito general o aplicado a sus asignaturas, que ha utilizado durante el cuatrimestre.
- Añadir nuevas estadísticas, como el tiempo de uso de los equipos, etc. Para desarrollarlo es necesario ampliar un sistema de gestión basado en bases de datos remotas.
- Permitir el bloqueo remoto de acceso a aulas determinadas, ya sea porque en el interior se estén solucionando problemas técnicos, porque se esté impartiendo alguna clase o porque el aula haya sido reservada.
- Implementar una funcionalidad que permita la utilización multiusuario de los equipos. Entendiendo como multiusuario aquellos casos en los que varios usuarios utilicen el mismo equipo para la realización, por ejemplo, de una práctica o trabajo común.
- No sólo restringir el uso del equipo a los usuarios permitidos sino también, y una vez iniciada la sesión en el PC, restringir el acceso a las aplicaciones informáticas, herramientas CAD, etc. que estén relacionadas con las asignaturas que está cursando el alumno.
- Incluir actividades de evaluación remota, autoevaluación, entrega de cuestionarios y/o prácticas, foros de debate *on-line*, teleaprendizaje, etc.

10. Referencias

10.1. Libros

[1] The Bologna Declaration of 19 June 1999, The European Higher Education Area.

[2] Jacobson, Ivar; Booch, Grady; Rumbaugh, James. "UML: el proceso unificado de desarrollo de software". Ed. AdissonWesley, 2000.

[3] Joyanes, Luis. "Programación Orientada a Objetos". Ed. McGrawHill, 1998.

[4] Moldes, F. Javier. "Java SE 6". Anaya Multimedia, 2008.

[5] Barclay, Kenneth y Savage, John. Object-Oriented Design With UML And Java. ButterWorth-Heinemann, 2003.

[6] Fuster Sabater, Amparo. Técnicas Criptográficas De Protección De Datos. 3ª Edición Actualizada. Editorial Ra-ma, 2009.

[7] Moldes Teo, Francisco Javier. Lenguaje C. 2ª Edición. Editorial ANAYA MULTIMEDIA, 2001.

10.2. Documentos PDF

[8] Sánchez Arriazu, Jorge. Descripción del algoritmo DES (Data Encryption Standard) , 1999. http://www.tierradelazaro.com/public/libros/des.pdf

[9] Garcia de Jalon de la Fuente, Javier. Aprenda lenguaje ANSI C como si estuviera en Primero, 1998.

http://www.tecnun.es/asignaturas/Informat1/ayudainf/aprendainf/AnsiC/leng_c.pdf

[10] LPC2468 User's Manual, version 2. www.embeddedartists.com

[11] Esquemáticos de la tarjeta LPC2468-16 OEM Board, versión 2. http://www.embeddedartists.com

[12] Documento del esquema de memoria (Memory Layout). http://www.embeddedartists.com

[13] Documento de especificación del diseño mecánico (OEM Board Mechanical Drawing). http://www.embeddedartists.com

10.3. URL's de interés

[14] http://java.sun.com/j2se/1.5.0/docs/api/

[15] http://www.embeddedartists.com

[16] http://www.nxp.com

[17] http://jusb.sourceforge.net/

[18] http://es.wikipedia.org/wiki/Declaración_de_Bolonia#cite_note-7

[19] http://es.wikipedia.org/wiki/Bus_de_Serie_Universal

[20] http://es.wikipedia.org/wiki/Disco_USB

Glosario

API – Aplication Programming Interface

ATA – Advanced Technology Attachment

CALUS – Control de Acceso Mediante Llave USB

CRL – Certificate Revocation List

DES – Data Encryption Standard

DSA – Digital Signature Algorithm

E2PROM – Electrically-Erasable Programmable Read-Only Memory

EA – Embedded Artists

ECB – Electronic Code Book

ECTS - European Credit Transfer System

EEES – Espacio Europeo de Educación Superior

EEPROM – Electrically-Erasable Programmable Read-Only Memory

ETM – Extended Text Message

FTP – File Transfer Protocol

IDE – Integrated Drive Electronics

J2EE – Java 2 Enterprise Edition

JAR – Java ARchive

JCA – J2EE Connector Architecture

JCA – Java Cryptography Architecture

JCE – Java Cryptography Extension

JDK – Java Development Kit

JMF – Java Media Framework

JTAG – Joint Test Action Group

JUSB – Java Universal Serial Bus

LCD – Licuid Crystal Display

LED – Light Emitter Diode

MAC – Message Autentication Code

MD5 – Message-Digest Algorithm 5

MMC/SD – MultiMedia Card/Secure Digital

NFS – Network File System

NIA – Número de Identificación de Alumno

OTG – On The Go

PAS – Personal de Administración y Servicios

PCB – Printed Circuit Board

PDI – Personal Docente y de Investigación

PKCS5Padding – Password-Based Cryptography Standard 5 Padding

PWM – Pulse Wide Modulation

QVGA – Quarter Video Graphics Array

RAM – Random Access Memory

RGB – Red Green Blue

RMI – (Java) Remote Method Invocation

ROM – Read-Only Memory

RSA – Rivest-Shamir-Adleman (Iniciales de los creadores del algoritmo: Ron Rivest, Adi Shamir y Len Adleman)

RTC – Real Time Clock

SCSI – Small Computer System Interface

SDK – Software Development Kit

SDRAM – Single Data Rate Synchronous Dynamic Random Access Memory

SHA – Secure Hash Algorithm

SPI – Serial Peripheral Interface

SVN – Subversion

TFT – Thin Film Transistor

TIC - tecnologías de la información y la comunicación

TripleDES – Triple Data Encryption Standard

UART – Universal Asynchronous Receiver-Transmitter

UML – Unified Modeling Language

USB – Universal Serial Bus

ANEXO 1. Presupuesto del Proyecto

A continuación se especifica el presupuesto que se ha invertido en la implementación del proyecto.

MATERIAL

Concepto	Precio (unidad)
LPC2468 OEM Board	239 €
OEM Base Board	
LCD con pantalla táctil QVGA de 3.2 pulgadas	
Tarjeta de memoria MMC/SD (2GB)	14 €
Lápiz para pantalla táctil	10 €
Memoria Flash USB (4GB)	12 €
TOTAL	**275 €**

Tabla 11.. Tabla con los precios del presupuesto.

OTROS COSTES

1. PC
2. Edificio... (10%)
3. Ingeniero (4 meses a 8 h/día x 90 euro/hora)

ANEXO 2. Términos criptográficos

Criptología – Es el estudio de la *criptografía* y el *criptoanálisis*.

Criptografía –En términos sociales, es la ciencia de hacer que el coste de adquirir o alterar información de modo impropio sea mayor que el posible valor obtenido al hacerlo. Desde el punto de vista más formal, es la práctica y el estudio de técnicas de *encriptación* y *desencriptación* de información, es decir, de técnicas para codificar un mensaje haciéndolo ininteligible (*encriptación*) y recuperar el mensaje original a partir de esa versión ininteligible (*desencriptación*).

Algoritmo criptográfico – Es un método matemático que se emplea para *encriptar* y *desencriptar* un mensaje. Generalmente funciona empleando una o más *claves* (números o cadenas de caracteres) como parámetros del algoritmo, de modo que sean necesarias para recuperar el mensaje a partir de la versión cifrada. El mensaje antes de encriptar se denomina *texto en claro* y una vez encriptado se denomina *texto cifrado*.

Sistema criptográfico – Es un sistema para *encriptar* y *desencriptar* información compuesto por un conjunto de *algoritmos criptográficos*, *claves* y, posiblemente, varios *textos en claro* con sus correspondientes versiones en *texto cifrado*. Los sistemas criptográficos actuales se basan en tres tipos de algoritmos criptográficos: simétricos o de clave secreta, de asimétricos o clave pública y de resumen de mensajes (funciones de dispersión).

Algoritmos de resumen de mensajes – Transforman mensajes de tamaño variable a textos cifrados de tamaño fijo sin emplear claves. Se emplean para convertir mensajes grandes en representaciones más manejables.

Algoritmos de clave secreta o simétricos – Convierten un mensaje en un texto cifrado del mismo tamaño que el original. Emplean una sola clave para encriptar y desencriptar. Son los algoritmos empleados para transferir grandes cantidades de información de modo seguro.

Algoritmos de clave pública o asimétricos – Encriptar un mensaje generando un texto cifrado del mismo tamaño que el original. Usan una clave para encriptar el mensaje (clave privada) y otra para desencriptar (clave pública). Tienen un coste computacional alto y se suelen emplear para distribuir las claves de los algoritmos simétricos.

Criptoanálisis – Es el conjunto de procedimientos, procesos y métodos empleados para romper un *algoritmo criptográfico*, *desencriptar* un *texto cifrado* o descubrir las *claves* empleadas para generarlo.